Computation of Green's functions for Boundary Value Problems with *Mathematica*[*]

Alberto Cabada[1], José Ángel Cid[2] and Beatriz Máquez–Villamarín[1]

July 13, 2012

1 Departamento de Análise Matemática, Facultade de Matemáticas, Universidade de Santiago de Compostela, 15782, Santiago de Compostela, Spain,
E-mail: alberto.cabada@usc.es, beatriz.maquez@usc.es
2 Departamento de Matemáticas, Universidade de Vigo, Higher Technical School of Computer Engineering, 32004, Ourense, Spain,
E-mail: angelcid@uvigo.es

Abstract

This paper is devoted to construct an algorithm that allows us to calculate the explicit expression of the Green's function related to a n^{th} – order linear ordinary differential equation, with constant coefficients, coupled with two – point linear boundary conditions. We develop this algorithm by making a *Mathematica* package.

Key Words: Green's function, boundary value problem, *Mathematica* package

1 Introduction

When we consider a n^{th} – order homogeneous linear ordinary differential equation, $L\,u(t) = 0$, $t \in [a, b]$, it is very well known that there are n linearly independent functions, which generate the general solution of the equation, i.e., any solution of the considered equation is a linear combination of such functions. When a inhomogeneous equation $L\,u(t) = \sigma(t)$, $t \in [a, b]$, is considered, we need to find a particular solution, for each σ given, and add it to the general expression of the homogeneous equation, and so we have the general solution of the inhomogeneous equation.

If we fix the values of u and its first $n - 1$ derivatives at the starting point of the interval a, under suitable regular assumptions on the data, we know that such initial problem is uniquely solvable. Moreover the expression of its unique solution is attained by obtaining the unique values of the coefficients in the expression given for the general case. When the coefficients of the linear operator are constants, such values are calculated by solving a related n^{th} – dimensional linear algebraic system.

However, when we treat problems in which the function and/or some of its derivatives up to order $n-1$, attain values at the two extremal points of the interval a and b, we have that the existence of solution of the linear two – point boundary value problem is, in general, not ensured. For this reason it is very important to develop tools that allow us to ensure the existence and uniqueness of the solution of the studied problem and, moreover, to calculate its exact expression.

[*]Partially supported by FEDER and Ministerio de Educación y Ciencia, Spain, project MTM2010-15314.

There are several methods to solve a boundary value problem, such as series expansions, the Laplace transform or the invariant imbedding (see [33]), but in our opinion the most appropriate way is by calculating the so called Green's function[1]: roughly speaking, if problem $L\,u = \sigma$, coupled with homogeneous linear boundary value conditions, has only the trivial solution for $\sigma \equiv 0$, then the associated linear operator is invertible and its inverse operator, $L^{-1}\sigma$, is characterized by an integral kernel, $G(t, s)$, called the Green's function. The solution of the problem $L\,u = \sigma$ is then given by

$$u(t) = L^{-1}\sigma(t) := \int_a^b G(t, s)\sigma(s)ds, \quad t \in [a, b].$$

We notice that, once we read the expression of the Green's kernel, we know the cases (if the linear operator depends on some parameters, for instance) in which it is not defined and, in consequence, the resonant cases (of non uniqueness of the homogeneous problem) are explicitly given. The main advantage of the Green's function is the fact that it is independent on the function σ. To get the exact solution for each particular case of σ we only need to calculate the corresponding integral, and so we have the expression that we are looking for. We do not need to develop a new computation for each σ: once we have the expression of function G, the problem is solved for any σ for which the integral is well defined (this is due to the fact that the Green's function is just the kernel of the operator L^{-1}).

Moreover, by means of the integral expression, we can also obtain some additional qualitative information about the solutions of the considered problem, such as their sign, oscillation properties, a priori bounds or their stability. For these reasons the theory of Green's functions is a fundamental tool in the analysis of differential equations. It has been widely studied in the literature [9, 15, 17, 18, 21, 32, 34, 36, 39, 40] and it has a great importance in order to use monotone iterative techniques, [24, 22], lower and upper solutions, [4, 6, 14, 24], fixed point theorems, [12, 36, 37, 38], or variational methods [35].

On the other hand, the explicit expression of the Green's function is, in general, very complicated and the calculations are carried out in a great technical difficulty. That is why our main goal consists on providing a software for the computation of the Green's function for a n^{th} – order linear problem with constant coefficients. To the best of our knowledge, this software is not available in the literature (see for instance [1, 20, 23, 25]).

Finally we remark that, from the physical point of view, the Green's function can be also defined as the response of a linear system to the unit impulse at time s, that is,

$$G(t, s) = L^{-1}\delta_s(t),$$

where δ_s is the unit Dirac measure concentrated at $t = s$ (see [18]).

The chapter is organized as follows: in the second section we present the basic facts of the general theory about existence and uniqueness of a Green's function for n^{th} – order linear problems coupled with linear boundary conditions. In the third section we focus our attention on n^{th} – order linear equations with constant coefficients and we develop and algorithm, based on some results in [6, 7, 8], to perform the computation of the Green's function by solving a $n \times n$ algebraic linear system. In the fourth section we review the relation between the sign of the Green's function and the maximum principles and its importance to develop monotone methods under the presence of lower and upper solutions. Finally, in the last section we use the scientific software *Mathematica* to implement the algorithm previously developed to effectively compute the Green's function. The entire code of the constructed package is showed, as an appendix, at the end of the paper.

[1] George Green (1793-1841) was the first mathematician to use such kind of kernels to solve boundary value problems.

2 Green's function: definition, existence and uniqueness

There are many books where the reader can consult the basic theory of Green's functions (see for instance [2, 5, 13, 18, 19, 28]). We shall follow the exposition in [28] by the good balance between clarity and generality.

We will consider two – point n^{th} – order linear boundary value problems of the form

$$L_n y(t) = \sigma(t),\ t \in I \equiv [a,b], \quad U_i(y) = \gamma_i,\ i = 1,\dots,m, \tag{1}$$

where

$$L_n y(t) \equiv a_0(t)\, y^{(n)}(t) + a_1(t)\, y^{(n-1)}(t) + \ldots + a_{n-1}(t)\, y'(t) + a_n(t)\, y(t), \quad t \in I,$$

and

$$U_i(y) \equiv \sum_{j=0}^{n-1} \left(\alpha_j^i\, y^{(j)}(a) + \beta_j^i\, y^{(j)}(b) \right), \qquad i = 1,\dots,m, \quad m \le n, \tag{2}$$

being α_j^i, β_j^i and γ_i real constants for all $i = 1,\dots,m$, and $j = 0,\dots,n-1$, σ and a_k continuous real functions for all $k = 0,\dots,n$, and $a_0(t) \neq 0$ for all $t \in I$.

It is well-known that the solution set of the linear equation $L_n y = 0$ is a n-dimensional vector subspace of $\mathcal{C}^n(I)$. Any base of such subspace will be called a *fundamental set of solutions* of equation $L_n y = 0$.

The homogeneous problem

$$L_n y(t) = 0,\ t \in I, \quad U_i(y) = 0,\ i = 1,\dots,m, \tag{3}$$

is said to be *k-compatible*, $0 \le k \le n$, if its set of solutions has dimension equals to k.

The following propositions are simple exercises of Linear Algebra.

Proposition 2.1. *Let $(y_1,\dots,y_n)$ be a fundamental set of solutions of $L_n y = 0$. Then, problem (3) is k-compatible if and only if*

$$\mathit{rank} \begin{pmatrix} U_1(y_1) & \dots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_m(y_1) & \dots & U_m(y_n) \end{pmatrix} = n - k.$$

Corollary 2.1. *The homogeneous problem (3) has only the trivial solution if and only if*

$$\mathit{rank} \begin{pmatrix} U_1(y_1) & \dots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_m(y_1) & \dots & U_m(y_n) \end{pmatrix} = n.$$

In particular the number of boundary conditions, m, must be equal to the order of the equation, n.

Proposition 2.2. *Let $(y_1,\dots,y_n)$ be a fundamental set of solutions of $L_n y = 0$ and let y_p be an arbitrary solution of the complete equation $L_n y = \sigma$. Then, any translation of the solution set of problem (1) to the origin has dimension equals to k if and only if*

$$\mathit{rank} \begin{pmatrix} U_1(y_1) & \dots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_m(y_1) & \dots & U_m(y_n) \end{pmatrix} = \mathit{rank} \left(\begin{array}{ccc|c} U_1(y_1) & \dots & U_1(y_n) & \gamma_1 - U_1(y_p) \\ \vdots & \ddots & \vdots & \vdots \\ U_m(y_1) & \dots & U_m(y_n) & \gamma_n - U_m(y_p) \end{array}\right) = n-k.$$

As consequence of the previous propositions we obtain the celebrated Fredholm's Alternative Theorem.

Theorem 2.2 (Fredholm's Alternative). *The problem*

$$L_n y(t) = \sigma(t),\ t \in I, \quad U_i(y) = \gamma_i,\ i = 1, \dots, n, \tag{4}$$

where the number, n, of boundary conditions equals the order of the linear equation, has a unique solution if and only if the associated homogeneous problem

$$L_n y(t) = 0,\ t \in I, \quad U_i(y) = 0,\ i = 1, \dots, n, \tag{5}$$

has only the trivial solution.

Now we introduce the axiomatic definition for the Green's function associated to the problem (3).

Definition 2.1. We say that G is a *Green's function* for problem (3) if it satisfies the following properties:

(G1) G is defined on the square $I \times I$.

(G2) For $k = 0, 1, \dots, n-2$, the partial derivatives $\dfrac{\partial^k G}{\partial t^k}$ exist and they are continuous on $I \times I$.

(G3) $\dfrac{\partial^{n-1} G}{\partial t^{n-1}}$ and $\dfrac{\partial^n G}{\partial t^n}$ exist and are continuous on the triangles $a \le s < t \le b$ and $a \le t < s \le b$.

(G4) For each $t \in (a, b)$ there exist the lateral limits

$$\frac{\partial^{n-1} G}{\partial t^{n-1}}(t, t^+) \quad \text{and} \quad \frac{\partial^{n-1} G}{\partial t^{n-1}}(t, t^-)$$

(i.e., the limits when $(t, s) \to (t, t)$ with $s > t$ or with $s < t$) and, moreover

$$\frac{\partial^{n-1} G}{\partial t^{n-1}}(t, t^+) - \frac{\partial^{n-1} G}{\partial t^{n-1}}(t, t^-) = -\frac{1}{a_0(t)}.$$

(G5) For each $s \in (a, b)$, the function $t \to G(t, s)$ is a solution of the differential equation $L_n y = 0$ on $t \in [a, s)$ and $t \in (s, b]$. That is,

$$a_0(t)\, \frac{\partial^n G}{\partial t^n}(t, s) + a_1(t)\, \frac{\partial^{n-1} G}{\partial t^{n-1}}(t, s) + \dots + a_{n-1}(t)\, \frac{\partial G}{\partial t}(t, s) + a_n(t)\, G(t, s) = 0,$$

on both intervals.

(G6) For each $s \in (a, b)$, the function $t \to G(t, s)$ satisfies the boundary conditions $U_i(G(\cdot, s)) = 0$, $i = 1, \dots, m$:

$$\sum_{j=0}^{n-1} \left(\alpha_j^i\, \frac{\partial^j G}{\partial t^j}(a, s) + \beta_j^i\, \frac{\partial^j G}{\partial t^j}(b, s) \right) = 0, \qquad i = 1, \dots, m.$$

The main importance of the previous definition is that the integral operator, whose kernel is a Green's function, gives us the solution for the semi-homogeneous problem

$$L_n y(t) = \sigma(t),\ t \in I, \quad U_i(y) = 0,\ i = 1, \dots, m, \tag{6}$$

as it is showed in the following result.

Theorem 2.3. *Let G be a Green's function of problem (3). Then, for each continuous function σ, we have that*

$$y(t) = \int_a^b G(t, s)\, \sigma(s)\, ds, \quad t \in I, \tag{7}$$

is a solution of problem (6).

The following is the main result of this section. We present the proof because it is constructive.

Theorem 2.4. *Let us suppose that the homogeneous problem (5) has only the trivial solution. Then there exists a unique Green's function, $G(t,s)$, related to (5). Moreover, for each continuous function σ, the unique solution of problem*

$$L_n y(t) = \sigma(t),\ t \in I, \quad U_i(y) = 0,\ i = 1, \dots, n, \tag{8}$$

is given by expression (7).

Proof. First we will show that a Green's function related to the initial value problem

$$L_n y(t) = 0,\ t \in I, \quad y^{(i)}(a) = 0,\ i = 0, \dots, n-1, \tag{9}$$

is given by

$$\tilde{K}(t,s) = \begin{cases} K(t,s), & \text{if} \quad a \le s \le t, \\ \\ 0, & \text{if} \quad t < s \le b, \end{cases}$$

where

$$K(t,s) := \frac{\begin{vmatrix} y_1(s) & \dots & y_n(s) \\ y_1'(s) & \dots & y_n'(s) \\ \vdots & \ddots & \vdots \\ y_1^{(n-2)}(s) & \dots & y_n^{(n-2)}(s) \\ y_1(t) & \dots & y_n(t) \end{vmatrix}}{a_0(s)\, W(y_1, \dots, y_n)(s)},$$

being $(y_1, \dots, y_n)$ a fundamental set of solutions of equation $L_n y = 0$, and

$$W(y_1, \dots, y_n)(s) = \begin{vmatrix} y_1(s) & \dots & y_n(s) \\ y_1'(s) & \dots & y_n'(s) \\ \vdots & \ddots & \vdots \\ y_1^{(n-1)}(s) & \dots & y_n^{(n-1)}(s) \end{vmatrix}$$

is its corresponding Wronskian.

Following Definition 2.1, we only need to check that function $\tilde{K}$ satisfies properties (G1) – (G6).

It is not difficult to verify that, for $k = 0, 1, \dots, n-2$, the partial derivatives $\partial^k \tilde{K} / \partial t^k$ exist and they are equal to

$$\frac{\partial^k \tilde{K}}{\partial t^k}(t,s) = \begin{cases} \dfrac{\begin{vmatrix} y_1(s) & \dots & y_n(s) \\ y_1'(s) & \dots & y_n'(s) \\ \vdots & \ddots & \vdots \\ y_1^{(n-2)}(s) & \dots & y_n^{(n-2)}(s) \\ y_1^{(k)}(t) & \dots & y_n^{(k)}(t) \end{vmatrix}}{a_0(s)\, W(y_1, \dots, y_n)(s)}, & \text{if} \quad a \le s \le t, \\ \\ 0, & \text{if} \quad t < s \le b, \end{cases}$$

so, they are continuous on $I \times I$ for all $n \ge 2$ (notice that if $s = t$ then two rows are equal in the determinant) . As consequence, condition (G2) is satisfied.

The same formula is also valid on the two triangles for the partial derivatives

$$\frac{\partial^{n-1} \tilde{K}}{\partial t^{n-1}}(t,s) \quad \text{and} \quad \frac{\partial^n \tilde{K}}{\partial t^n}(t,s),$$

so condition (G3) is also satisfied.

With respect to the lateral limits on the previous derivative, we have that

$$\frac{\partial^{n-1}\tilde{K}}{\partial t^{n-1}}(t,t^{+}) = 0,$$

and

$$\frac{\partial^{n-1}\tilde{K}}{\partial t^{n-1}}(t,t^{-}) = \frac{\partial^{n-1}K}{\partial t^{n-1}}(t,t^{-}) = \frac{\begin{vmatrix} y_1(t) & \dots & y_n(t) \\ y_1'(t) & \dots & y_n'(t) \\ \vdots & \ddots & \vdots \\ y_1^{(n-2)}(t) & \dots & y_n^{(n-2)}(t) \\ y_1^{(n-1)}(t) & \dots & y_n^{(n-1)}(t) \end{vmatrix}}{a_0(t)\,W(y_1,\dots,y_n)(t)} = \frac{1}{a_0(t)}.$$

So, we conclude that condition (G4) is fulfilled.

On the first triangle, $a \le s < t$, for fixed s, $\tilde{K}(t,s)$ is a linear combination of $y_1(t),\dots,y_n(t)$, so condition (G5) holds.

Obviously, for all $s \in (a,b)$ it is satisfied that

$$\frac{\partial^k}{\partial t^k}\tilde{K}(a,s) = 0 \qquad \text{for all} \quad k = 0,\dots,n-1.$$

Therefore $\tilde{K}(t,s)$ satisfies conditions (G1) – (G6) with respect to the initial value problem (9).

Now, in order to construct a Green's function for problem (5), let us consider n continuous functions on I, $c_1,\dots,c_n$. Now we are going to look for a Green's function of the form

$$G(t,s) = \tilde{K}(t,s) + c_1(s)\,y_1(t) + \ldots + c_n(s)\,y_n(t).$$

It is easy to verify that function G satisfies conditions (G1) – (G5). It only rests to show that it is possible to choose $c_1,\dots,c_n$ such that G satisfies (G6) for problem (5), that is, for each $s \in (a,b)$ we need to verify that

$$U_i(G(\cdot,s)) = 0, \quad \forall\, i = 1,\dots,n, \qquad \forall\, s \in I.$$

By linearity, we have that

$$U_i(G(\cdot,s)) = U_i(\tilde{K}(\cdot,s)) + \sum_{j=1}^{n} c_j(s)\,U_i(y_j), \quad i = 1,\dots,n,$$

that is, $(c_1(s),\dots,c_n(s))$ should be a solution of the linear system

$$\begin{pmatrix} U_1(y_1) & \dots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_n(y_1) & \dots & U_n(y_n) \end{pmatrix} \begin{pmatrix} c_1(s) \\ \vdots \\ c_n(s) \end{pmatrix} = -\begin{pmatrix} U_1(\tilde{K}(\cdot,s)) \\ \vdots \\ U_n(\tilde{K}(\cdot,s)) \end{pmatrix}.$$

Now, since the homogeneous boundary value problem (5) has only the trivial solution, we know from Corollary 2.1 that the rank of the previous matrix is equal to n. Thus, the system has a unique solution given by

$$\begin{pmatrix} c_1(s) \\ \vdots \\ c_n(s) \end{pmatrix} = -\begin{pmatrix} U_1(y_1) & \dots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_n(y_1) & \dots & U_n(y_n) \end{pmatrix}^{-1} \begin{pmatrix} U_1(\tilde{K}(\cdot,s)) \\ \vdots \\ U_n(\tilde{K}(\cdot,s)) \end{pmatrix}.$$

Moreover, from this expression, we know that functions $c_1,\dots,c_n$ are continuous and, therefore, G is a Green's function of problem (5).

Now, from Theorem 2.3, we know that

$$y(t) = \int_a^b G(t,s)\,\sigma(s)\,ds, \quad t \in I,$$

is a solution of problem (8) and, moreover, this solution is unique by Fredholm's Alternative Theorem (see Theorem 2.2).

Now, to prove the uniqueness of the Green's function, let us suppose that $\tilde{G}$ is another function satisfying conditions (G1) – (G6) for problem (5). Then for all $t \in I$ and all continuous function σ, it is satisfied that

$$\int_a^b G(t,s)\,\sigma(s)\,ds = \int_a^b \tilde{G}(t,s)\,\sigma(s)\,ds, \quad t \in I,$$

which implies that

$$G(t,s) = \tilde{G}(t,s),$$

for all $t, s \in I$, if $n \geq 2$, and for all $t \neq s$ if $n = 1$, because in this last case the Green's function is discontinuous at the diagonal of the square of definition. □ □

3 Computation of the Green's function for linear differential equations with real constant coefficients

In this section we deal with the two – point boundary value problem

$$\mathcal{L}_n y(t) = \sigma(t),\ t \in I, \quad U_i(y) = 0,\ i = 1, \ldots, n, \tag{10}$$

where now the linear equation has real constant coefficients, that is:

$$\mathcal{L}_n y(t) = y^{(n)}(t) + a_1\, y^{(n-1)}(t) + \ldots + a_{n-1}\, y'(t) + a_n\, y(t), \quad t \in I,$$

with $a_1, a_2, \ldots, a_n \in \mathbb{R}$. Moreover, let us suppose that the linear functionals U_i defined by (2), with $a_j, \alpha_j^i, \beta_j^i \in \mathbb{R}$ for all $i = 1, \ldots, n$ and $j = 0, \ldots, n-1$, are such that the asociated homogeneous problem

$$\mathcal{L}_n y(t) = 0,\ t \in I, \quad U_i(y) = 0,\ i = 1, \ldots, n, \tag{11}$$

has only the trivial solution.

Clearly both problems are particular cases of (8) and (5), respectively.

We will develop a method to compute explicitly the Green's function. Such a method will allow us to obtain its expression by solving a linear algebraic system of dimension n. The following result for initial value problems is proved in [8].

Theorem 3.1. *Let r be the unique solution of the initial value problem*

$$\begin{aligned} u^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i}\, u^{(i)}(t) = 0, \quad t \in \mathbb{R}, \\ u^{(i)}(0) = 0, \quad i = 0, \ldots, n-2, \\ u^{(n-1)}(0) = 1. \end{aligned} \tag{12}$$

Then, the unique solution of the initial value problem

$$\begin{aligned} y^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i}\, y^{(i)}(t) = \sigma(t), \quad t \in I, \\ y^{(i)}(a) = \lambda_i, \quad i = 0, \ldots, n-1, \end{aligned} \tag{13}$$

with $\sigma \in \mathcal{C}(I)$ *and* $\lambda_i \in \mathbb{R}$, $i = 0, \ldots, n-1$, *is given by*

$$y(t) = \int_a^t r(t-s)\,\sigma(s)\,ds + \sum_{k=0}^{n-1} y_k(t)\,\lambda_k, \tag{14}$$

where

$$y_k(t) = r^{(n-1-k)}(t-a) + \sum_{j=k+1}^{n-1} a_{n-j}\,r^{(j-k-1)}(t-a), \quad t \in \mathbb{R}, \quad k = 0, \ldots, n-1. \tag{15}$$

Proof. Let us define

$$v(t) = \int_a^t r(t-s)\,\sigma(s)\,ds, \quad t \in I.$$

From the definition of r and the differentiation Leibniz's rule, we obtain that

$$v^{(i)}(t) = \int_a^t r^{(i)}(t-s)\,\sigma(s)\,ds, \quad t \in I, \qquad i = 0, 1, \ldots, n-1,$$

and

$$v^{(n)}(t) = \int_a^t r^{(n)}(t-s)\,\sigma(s)\,ds + \sigma(t), \quad t \in I.$$

In consequence,

$$v^{(i)}(a) = 0, \quad t \in I \qquad \forall\, i = 0, 1, \ldots, n-1$$

and, for all $t \in I$,

$$v^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i}\,v^{(i)}(t) = \int_a^t \Big[r^{(n)}(t-s) + \sum_{i=0}^{n-1} a_{n-i}\,r^{(i)}(t-s)\Big]\,\sigma(s)\,ds + \sigma(t) = \sigma(t).$$

On the other hand, it is clear that

$$\begin{aligned}(r')^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i}\,(r')^{(i)}(t) &= 0, \quad t \in \mathbb{R},\\ r'(0) = \ldots = r^{(n-2)}(0) &= 0\\ r^{(n-1)}(0) &= 1\\ r^{(n)}(0) &= -a_1.\end{aligned}$$

So, the function $y_{n-2}(\cdot) \equiv r'(\cdot - a) + a_1\,r(\cdot - a)$ is the unique solution of problem

$$\begin{aligned}y^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i}\,y^{(i)}(t) = 0, &\quad t \in \mathbb{R},\\ y^{(i)}(a) = 0, &\quad i = 0, \ldots, n-1,\, i \neq n-2,\\ y^{(n-2)}(a) = 1.&\end{aligned}$$

Analogously, we can prove that for all $j \in \{0, 1, \ldots, n-1\}$, the function y_{n-1-j} is the unique solution of

$$y^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i}\,y^{(i)}(t) = 0, \quad t \in \mathbb{R}, \tag{16}$$

$$y^{(i)}(a) = 0, \quad i = 0, \ldots, n-1,\, i \neq n-1-j, \tag{17}$$

$$y^{(n-1-j)}(a) = 1. \tag{18}$$

Therefore,

$$u(t) = v(t) + \sum_{i=0}^{n-1} y_i(t)\,\lambda_i, \quad t \in I,$$

is the unique solution of problem (13), as we want to prove. □ □

From the previous result, we know that the unique solution of the initial problem (13) is given by the expression (14), where r is the unique solution of problem (12), the functions y_k satisfy (15) and λ_k are given real parameters.

Analogously to the proof of Theorem 2.4, if we consider the bounday value problem (10), we will search for a Green's function of the form

$$G(t,s) = \begin{cases} r(t-s) + \sum\limits_{k=0}^{n-1} y_k(t)\,d_k(s), & \text{if} \quad a \le s \le t \le b, \\ \sum\limits_{k=0}^{n-1} y_k(t)\,d_k(s), & \text{if} \quad a \le t < s \le b, \end{cases} \tag{19}$$

where the continuous real functions d_k are the unknowns.

Taking into account (7) and (19), we obtain

$$\begin{aligned} 0 &= \sum_{j=0}^{n-1} \Big(\alpha_j^i\, y^{(j)}(a) + \beta_j^i\, y^{(j)}(b) \Big) \\ &= \sum_{j=0}^{n-1} \Big[\alpha_j^i \int_a^b \sum_{k=0}^{n-1} y_k^{(j)}(a)\, d_k(s)\,\sigma(s)\,ds + \\ &\quad + \beta_j^i \int_a^b r^{(j)}(b-s)\,\sigma(s)\,ds + \beta_j^i \int_a^b \sum_{k=0}^{n-1} y_k^{(j)}(b)\,d_k(s)\,\sigma(s)\,ds \Big] \\ &= \sum_{j=0}^{n-1} \Big[\beta_j^i \int_a^b r^{(j)}(b-s)\,\sigma(s)\,ds \Big] + \sum_{j=0}^{n-1} \int_a^b d_k(s) \Big[\alpha_j^i \sum_{k=0}^{n-1} y_k^{(j)}(a) + \beta_j^i \sum_{k=0}^{n-1} y_k^{(j)}(b) \Big] \sigma(s)\,ds \\ &= \int_a^b \left[\sum_{j=0}^{n-1} \beta_j^i\, r^{(j)}(b-s) + \sum_{k=0}^{n-1} d_k(s)\, U_i(y_k) \right] \sigma(s)\,ds. \end{aligned}$$

Since y_k, r and U_i has been previously obtained, by solving the linear system

$$\sum_{k=0}^{n-1} d_k(s)\, U_i(y_k) = -\sum_{j=0}^{n-1} \beta_j^i\, r^{(j)}(b-s), \qquad i = 1, \dots, n, \tag{20}$$

we obtain the expression of $d_k(s)$ and, therefore, we have the formula for $G(t,s)$.

Notice that system (20) is equivalent to

$$\begin{pmatrix} U_1(y_0) & \cdots & U_1(y_{n-1}) \\ \vdots & \ddots & \vdots \\ U_n(y_0) & \cdots & U_n(y_{n-1}) \end{pmatrix} \begin{pmatrix} d_0(s) \\ \vdots \\ d_{n-1}(s) \end{pmatrix} = - \begin{pmatrix} \sum\limits_{j=0}^{n-1} \beta_j^1\, r^{(j)}(b-s) \\ \vdots \\ \sum\limits_{j=0}^{n-1} \beta_j^n\, r^{(j)}(b-s) \end{pmatrix}. \tag{21}$$

Since problem (11) has only the trivial solution, from Theorem 2.2 and Corollary 2.1, we have that the previous system is uniquely solvable.

3.1 The periodic problem

As we have seen in the proof of Theorem 3.1, to obtain the expression of the Green's function related to problem (11) is reduced to solve the algebraic system (21). Moreover the existence and uniqueness of such function is equivalent to the uniqueness of solution of the corresponding algebraic system.

From the definition of the functions y_k given in (15), we only need to calculate the function r as the unique solution of the initial value problem (12). Despite this, in some cases, depending on the order of the equation and the nonzero parameters involved in the operator $\mathcal{L}_n$ and the functionals U_i, the expressions of the continuous functions d_k can present a very complicated expression which is very difficult to simplify. Is for this reason that in some particular cases of boundary conditions one can find more suitable ways to calculate the Green's function. Due to its importance, we make special mention to the periodic case, that is,

$$U_i(y) \equiv y^{(i)}(b) - y^{(i)}(a), \quad i = 0, \dots, n-1.$$

In this situation, the expression of the Green's function is showed in the following result, which has been proved in [7, Lemma 2.1] (see also [6, Lemma 2.1]) for the particular interval $[0, 2\pi]$. The proof follows similar steps to those of Theorem 3.1 and so it is omitted.

Lemma 3.2. *Suppose that the linear periodic boundary value problem:*

$$\mathcal{L}_n\, u(t) = \sigma(t),\ t \in I, \quad u^{(i)}(a) - u^{(i)}(b) = \mu_i, \quad i = 0, \dots, n-1,$$

has a unique solution $u \in \mathcal{C}^n(I)$ for all $\sigma \in \mathcal{C}(I)$.

Then u is given by the expression

$$u(t) = \int_a^b G(t,s)\, \sigma(s)\, ds + \sum_{i=0}^{n-1} r_i(t)\, \mu_i.$$

Here $r_j \in C^\infty(\mathbb{R})$, $j = 0, 1, \dots, n-1$, is the unique solution of the linear problem:

$$\begin{aligned} \mathcal{L}_n\, z(t) &= 0, \quad t \in \mathbb{R}, && (22)\\ z^{(i)}(a) - z^{(i)}(b) &= 0, \quad i = 0, \dots, n-1\ ;\ i \neq j, && (23)\\ z^{(j)}(a) - z^{(j)}(b) &= 1, && (24) \end{aligned}$$

and

$$G(t,s) = \begin{cases} r_{n-1}(a+t-s), & \text{if} \quad a \le s \le t \le b \\ r_{n-1}(b+t-s), & \text{if} \quad a \le t \le s \le b. \end{cases} \qquad (25)$$

Furthermore, for all $i = 0, \dots, n-2$, the following expression holds:

$$r_i(t) = r_{n-1}^{(n-1-i)}(t) + \sum_{j=1}^{n-i-1} a_{n-j}\, r_{n-1}^{(j-i-1)}(t), \quad t \in I.$$

It is obvious that the expression given in the previous lemma is easier to compute that the one given in the first part of this section. In this case we only need to obtain the expression of the function r_{n-1}, which is deduced by solving a $n \times n$ constant linear algebraic system.

4 Green's functions, maximum principles and monotone iterative methods

It is well know that if u is a twice differentiable function on a compact interval I and $u'' \geq 0$ on the interior of I then u satisfies a (strong) maximum principle, that is: if u attains its maximum in an interior point then u is a constant function on I.

Maximum principles, in their different forms, play a very important role in the study of differential equations (see [26, 31]). In particular the above maximum principle implies that whenever u satisfies

$$u''(t) \geq 0 \quad \text{for all } t \in \ I, \quad u(\partial I) \leq 0,$$

then $u \leq 0$ on I (in fact as the maximum principle is strong either $u \equiv 0$ or $u < 0$ on the interior of I).

This last result (also called in the literature maximum principle) is fundamental in order to implement monotone iterative methods. For instance, when dealing with the second order Dirichlet problem

$$u''(t) = f(t, u(t)), \ t \in I, \quad u(\partial I) = 0, \tag{26}$$

the maximum principle allows to prove the convergence of the iterative sequence

$$u''_{n+1}(t) = f(t, u_n(t)), \ t \in I, \quad u_{n+1}(\partial I) = 0,$$

$$u_0 = \alpha \quad \text{or} \quad u_0 = \beta,$$

to a solution of (26), being $\alpha \leq \beta$ lower and upper solutions ([14]), that is, they satisfy the differential inequalities

$$\alpha''(t) \geq f(t, \alpha(t)), \ t \in I, \quad \alpha(\partial I) \leq 0,$$

$$\beta''(t) \leq f(t, \beta(t)), \ t \in I, \quad \beta(\partial I) \geq 0.$$

In an abstract setting we say that operator L satisfies the maximum principle if

$$Lu \geq 0 \quad \text{on } I \implies u \leq 0 \quad \text{on } I,$$

and that L satisfies an anti-maximum principle if

$$Lu \geq 0 \quad \text{on } I \implies u \geq 0 \quad \text{on } I.$$

Concerning the general boundary value problem (6), we recall (see Section 2) that if the linear homogeneous boundary value problem (5) is non resonant (i.e., it has only the trivial solution) then the unique solution of the problem (6) is given by the expression (7).

As consequence, it is clear that if the linear operator L satisfies a maximum or an anti-maximum principle then the homogeneous problem is nonresonant and there exists L^{-1}. Then, the maximum principle is equivalent to L^{-1} be inverse negative (that is, $u \geq 0 \Rightarrow L^{-1}u \leq 0$) while the anti-maximum principle is equivalent to L^{-1} be inverse positive (that is, $u \geq 0 \Rightarrow L^{-1}u \geq 0$). So, in terms of the sign of the Green's function, the maximum principle is equivalent to $G(t, s) \leq 0$ for all $(t, s) \in I \times I$ and the anti-maximum principle is equivalent to $G(t, s) \geq 0$ for all $(t, s) \in I \times I$. For this reason the searching for conditions implying the constant sign of the Green's function has received a lot of attention from many researchers in the last decades (see [3, 9, 11, 21, 29, 30, 36, 39] and references therein).

5 Implementation of the Green's function by means of *Mathematica*

The purpose of this section is to construct an algorithm for computing the Green's function of problem (10). Such construction is based on the expression (19). To arrive at such expression, we must previously found the functions r, y_k and d_k.

Due to the fact that the function r is the unique solution of the initial value problem (12), the first step consists on solving such problem.

Once we have this expression we obtain the expression of y_k, the unique solutions of the related problems (16) – (18).

The next step of the algorithm is to solve the system (20). In consequence, to ensure the existence and uniqueness of the Green's function, we must verify, in a first moment, that the matrix of system (20) is invertible. Otherwise, there is not Green's function and this ends the process.

When the system (20) is uniquely solvable, once we have obtained its unique solution, d_k, we arrive at the expression of the Green's function, $G(t,s)$, by means of the expression (19) defined in the two triangles $a \leq s < t \leq b$ and $a \leq t < s \leq b$.

As we have noticed in the previous section, the calculations involved in this process are very complicated, so, for higher order equations and several boundary conditions, the resolution may be very slow and the simplifications unavailable. On the other hand, for the particular case of homogeneous periodic boundary conditions we only must obtain the function r_{n-1}, defined in (22) – (24), in order to have the expression of the Green's function from equation (25).

The outline of the described algorithm can be seen in the flow diagram of Figure 1.

5.1 The *Module* enviroment to calculate the Green's Function

By using the scientific software *Mathematica 8.0.1.0* and the previously described algorithm, we implemented a program in which, by supplying the order equation, the coefficients of the linear operator and the two – point boundary conditions on the interval I, it is calculated the related Green's function.

Using the *Manipulate* environment, and taking advantage of the *Module* that we are going to describe, we will design a "friendly" environment that allows to the user to enter data and interact with the program in a simple manner. Moreover, we can see and manipulate both the analytical outputs as the graphical results.

In this section we focus our attention on the technical aspects of the *Module* and we leave the aspects of the *Manipulate* to the next subsection.

The *Mathematica Module* that we are going to describe, runs once we previously know the following values:

1. the order of the equation (**n**),
2. the coefficients (**vector c**),
3. the extremes of the interval (**exta** and **extb**)
4. the two – point boundary conditions depending on a and b (**vector cc**).

Moreover we have the option **Periodicity** (False / True).

As consecuence we can enter the values for the aforementioned variables and simply pick up at the *Module* to run. As we have noticed, in the next section we will explain how to insert this *Module* into the *Manipulate* to enhance the user interaction.

Our *Module* is divided into two algorithms, depending on whether the boundary conditions are, or not, the periodic ones. In this last case, we can choose the more

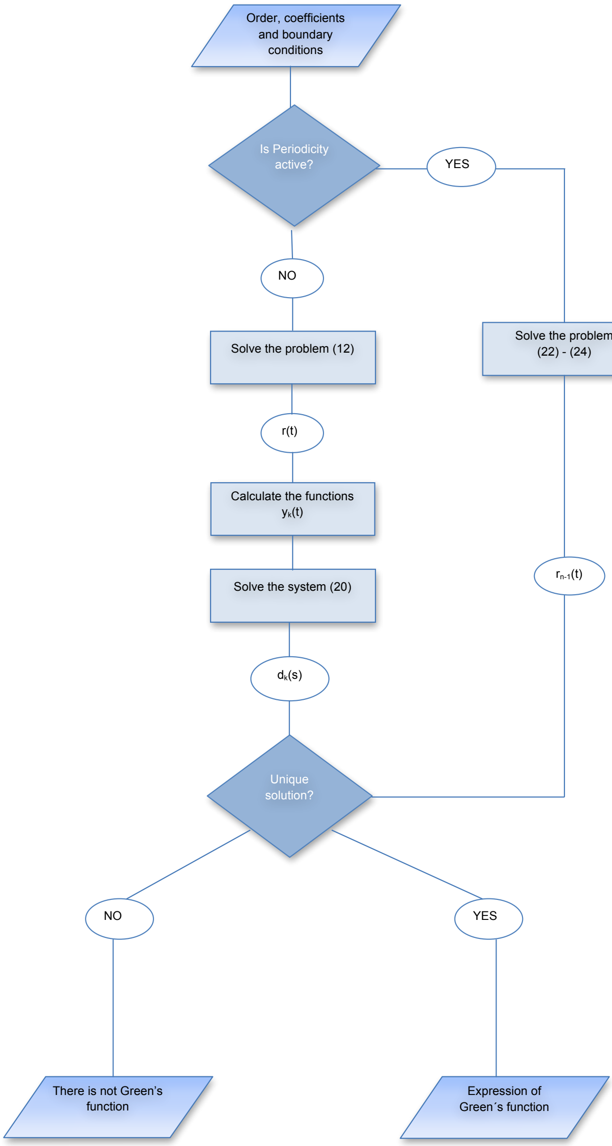

Figure 1: Flow diagram of the algorithm.

specific algorithm explained at the beggining of this section. Let us start to describe the generic algorithm presented in the first part of section 5, which will become if we keep the **Periodicity** option as False.

At the beginning, the program checks that both the number of the coefficients and of the boundary conditions equal the order of the considered equation. Due to the fact that such conditions should be evaluated in the *Module*, in order to find the coefficients of the system described in (20), these boundary conditions must be a real function and not a vector, as they have been introduced by the keyboard. We have decided to ask for a n – dimensional vector because in this case it is not necessary to enter all the coefficients α_j^i, β_j^i, in functions U_i. It is the *Module* which calculates these coefficients directly from the introduced vector.

To carry out the transformation of each boundary condition into a real function, it is defined an auxiliary function at the beginning of the *Module* (**aux**). This function saves the vector that contains the boundary conditions as a function that depends on u, *exta* and *extb*. After transforming it into a system, the coefficients that multiply $u^{(j)}(exta)$ and $u^{(j)}(extb)$ are extracted as α_j^i and β_j^i, respectively. So the function is well defined.

The part of the code that makes this step is showed below. Here **aux** is the auxiliary function, that has been previously defined from the vector that contains the boundary conditions:

$$\texttt{Do}\left[\texttt{alfa}[i,j]=\texttt{Coefficient}\left[\texttt{aux}[u][exta,extb][[i]],u^{(j)}[exta]\right],\{j,0,\texttt{Length}[c]-1\},\{i,1,\texttt{Length}[c]\}\right];$$

$$\texttt{Do}\left[\texttt{beta}[i,j]=\texttt{Coefficient}\left[\texttt{aux}[u][exta,extb][[i]],u^{(j)}[extb]\right],\{j,0,\texttt{Length}[c]-1\},\{i,1,\texttt{Length}[c]\}\right];$$

Once the coefficients α_j^i and β_j^i have been extracted, are defined the functionals U_i, which depends on u, *exta* and *extb*:

$$\texttt{Do}\left[U_i[\texttt{u\_}][\texttt{exta\_},\texttt{extb\_}]=\sum_{j=0}^{\texttt{Length}[c]-1}\left(\texttt{alfa}[i,j]*u^{(j)}[exta]+\texttt{beta}[i,j]*u^{(j)}[extb]\right),\{i,1,\texttt{Length}[c]\}\right]$$

The first step of the algorithm mentioned above is to find the unique solution of the initial value problem (12). This problem is solved by using the *Mathematica DSolve* command. The result is saved as a function r that depends on t.

Denoting by n the order of the equation and by c the vector where the coefficients are saved, the equation (12) is introduced in the program as:

$$\texttt{ec:=}y^{(\texttt{Length}[c])}[t]+\sum_{i=1}^{\texttt{Length}[c]}c[[i]]\,y^{(\texttt{Length}[c]-i)}[t].$$

The initial value problem (12) is solved in the sentence

$$\texttt{DSolve}\left[\texttt{Join}\left[\{\texttt{ec}==0\},\texttt{Table}\left[y^{(i)}[0]==0,\{i,0,n-2\}\right],\left\{y^{(n-1)}[0]==1\right\}\right],y,t\right];$$

Note that this output is a list, so we need to extract the corresponding part of the function with $y[t]/.\text{DSolve}[\cdots][[1]]$. The input *DSolve* returns the simplified solution, and so on numerous occasions this result shows an expression in which it appears complex numbers when this expression of the solution is the shortest one. We notice that, for higher order equations, it is very usual that *Mathematica* solves the previous equation with dependence on the roots of the characteristic polynomial. In this case the expression of the solution appears as a function of the (unknown for *Mathematica*) corresponding roots. This makes the expression of the Green's function impossible to process in practical situations. For this reason the program checks if the word *Root* appears in the expression and, if it is the case, it makes the transformation $c = N[c]$ over the coefficients. This fact implies that *Mathematica* considers such coefficients as a numerical approximation of the corresponding exact numbers, and it makes the next calculations for functions r, y_k and, as consequence, for the Green's function G, as numerical approximations too. Our experiments show us that the numerical error is around 10^{-15}.

The second step of the algorithm consists on solving the associated problem (16) – (18). The solution is obtained either directly by using *DSolve* or by means of the expression (15), depending if either all the introduced data are real numbers or there is some parameter. The implementation in *Mathematica* is as follows:

$$\text{If}[(\texttt{c} \in \text{Reals} \wedge \text{extb} \in \text{Reals} \wedge \text{exta} \in \text{Reals})===\text{True},$$

$$\text{Do}[\text{soluci}[\texttt{k}] = \text{DSolve}\left[\text{Join}\left[\left\{\texttt{y}^{(\text{Length}[\texttt{c}])}[\texttt{t}] + \sum_{\texttt{i}=1}^{\text{Length}[\texttt{c}]} \texttt{c}[[\texttt{i}]]\texttt{y}^{(\text{Length}[\texttt{c}]-\texttt{i})}[\texttt{t}] == \texttt{0}\right\},\right.\right.$$

$$\text{Table}\left[\texttt{y}^{(\texttt{i})}[\texttt{0}] == \texttt{0}, \{\texttt{i}, \texttt{0}, \texttt{k}-\texttt{2}\}\right], \left\{\texttt{y}^{(\texttt{k}-\texttt{1})}[\texttt{0}] == \texttt{1}\right\},$$

$$\left.\left.\text{Table}\left[\texttt{y}^{(\texttt{i})}[\texttt{0}] == \texttt{0}, \{\texttt{i}, \texttt{k}, \text{Length}[\texttt{c}]-\texttt{1}\}\right]\right], \texttt{y}, \texttt{t}\right];$$

$$\text{yk}[\texttt{k}][\text{t\_}] = \text{FullSimplify}[\text{ComplexExpand}[\texttt{y}[\texttt{t}]/.\text{Extract}[\text{soluci}[\texttt{k}], \{\texttt{1}, \texttt{1}\}]]];,$$

$$\{\texttt{k}, \texttt{1}, \text{Length}[\texttt{c}]\}],$$

$$\text{Do}\left[\text{yk}[\texttt{k}][\text{t\_}] = \texttt{r}^{(\text{Length}[\texttt{c}]-\texttt{k})}[\texttt{t}] + \sum_{\texttt{j}=\texttt{k}}^{\text{Length}[\texttt{c}]-\texttt{1}} \texttt{c}[[\text{Length}[\texttt{c}]-\texttt{j}]]\texttt{r}^{(\texttt{j}-\texttt{k})}[\texttt{t}];,\right.$$

$$\{\texttt{k}, \texttt{1}, \text{Length}[\texttt{c}]\}]]$$

This different choice follows from our experimental experience with *Mathematica*. We have noticed that, when all the variables are real constants, the use of formula (15) gives us bigger numerical errors than the direct resolution of (16) – (18), but (15) is more adequate if some real parameter is involved in the equations.

The coefficients of the system (20) are given by the boundary conditions U_i evaluated at y_k. To solve it, we use the *Mathematica* command *Solve*, and the unknowns variables are saved on the vector d.

$$\texttt{Solve}\left[\texttt{Table}\left[0 == \sum_{j=0}^{\texttt{Length}[c]-1} \texttt{beta}[i,j] * r^{(j)}[extb - s] + \sum_{j=1}^{\texttt{Length}[c]} d_j[s]U_i[y[j]][exta, extb], \{i, 1, \texttt{Length}[c]\}\right], \texttt{Table}\left[d_i[s], \{i, 1, \texttt{Length}[c]\}\right]\right]$$

We notice that the variables used to solve this system must to be local ones. This is due to the fact that any overlap in the value of the vector d implies that if the dimension of the equation is changed then the system is incorrectly solved.

Finally, by means of the performed calculations, the Green's function is defined as in expression (19). To make it, we need to extract the coefficients of the solution of the system (20) returned by *Mathematica*, which is given in a list form. In the following lines of code we present the extraction of such coefficients and the definition of the function h that corresponds to the summation part on both sides of the expression (19).

$$\texttt{coef:=Sort}[\texttt{Extract}[\texttt{ecuacion}, \{1\}]];$$

$$\texttt{Do}\left[e[i][\texttt{s\_}]\texttt{:=}d_i[s]\texttt{/.Extract}[\texttt{coef}, \{i\}];, \{i, 1, n\}\right];$$

$$h[\texttt{t\_}, \texttt{s\_}]\texttt{:=} \sum_{i=1}^{\texttt{Length}[c]} \texttt{Simplify}\left[e[i][s]\right] y[i][t];$$

The so called function $h(t,s)$ is the most complicated part of the expression of the Green's function and the simplification becomes harder to do.

We recall that this *Module* has two parts, the first one is the generic algorithm, described above, and the second part consists on a specific algorithm to solve equations with periodic boundary conditions. The option **Periodicity** makes the program to run in one way or in another.

In the specific algorithm for periodic boundary conditions it is only needed to solve a boundary value problem and to define the Green's function as in equation (25).

5.2 An environment based on *Manipulate*

Once the program has been implemented, the next step is to make a simple environment for the input of the data. To this end, we have programed a *Manipulate* package, which is an interactive environment where users must enter the data through boxes or menus.

When running the program it will appear an environment as in Figure 2.

In this environment the user must enter the order of the equation, n, that should be a natural number. Then it must be introduced the vector of coefficients, which must be written in *Mathematica* format, i.e., between keys and separated by commas. The vector $\{c_1, \ldots, c_n\}$ must have length n and it contains the coefficients accompanying $u^{n-1)}, \ldots, u$.

The coefficients could be real numbers as well as parameters. These parameters can be any lowercase letter that has not been previously used in the calculations (for instance: j,k,l,m,...). Notice that the parameter is not consider as a variable by the Green's function, so the Graph option does not run in this case.

In the next two boxes the user must insert the endpoints of the interval I. The second one must be strictly bigger than the first one. It is allowed to insert at the endpoints the values a and/or b in a generic way, but in this case the option Graph does not run again, because they are considered as parameters and not as variables.

In the last box are introduced the boundary conditions. They must be inserted as a $n-$ dimensional vector, and must take its values at the previously given endpoints a and b. If we choose the option **Periodicity** the considered boundary conditions are the periodic ones and the program calculate them with the alternative algorithm explained at the beginning of this section. Of course, the periodic boundary conditions can also be introduced in the corresponding box and the program will made the calculations in the generic way.

The **Enter** button controls the execution of the *Manipulate*, while the program is making calculations it appears pressed.

5.3 Final remarks

Because of some of the calculations can exceed the maximum execution time of the machine outputs, we can have an unsatisfactory output that does not return *Manipulate* to the initial situation. In such a case it will be necessary to abort the execution and relaunch the program evaluating again the code.

By making suitable simplifications the output given by the program can be substantially improved. For instance, some commands as **ComplexExpand**, **ExpToTrig** or **Simplify**, may help to get the expression of the Green's function without complex numbers. We remark that $G[t, s]$, where the Green's function is saved, is a global variable, so once the execution is completed the user can simplify it outside the *Module* by applying the commands that are more suitable in each case.

The complete code of the program is included at the end of the paper as an Appendix. Alternatively it can be downloaded from the web page

`http://webspersoais.usc.es/persoais/alberto.cabada/index.html`

Notice that *Mathematica 8.0.1.0* is recommended in order to run the program. Some troubles could arise if the program is executed in other versions of *Mathematica*.

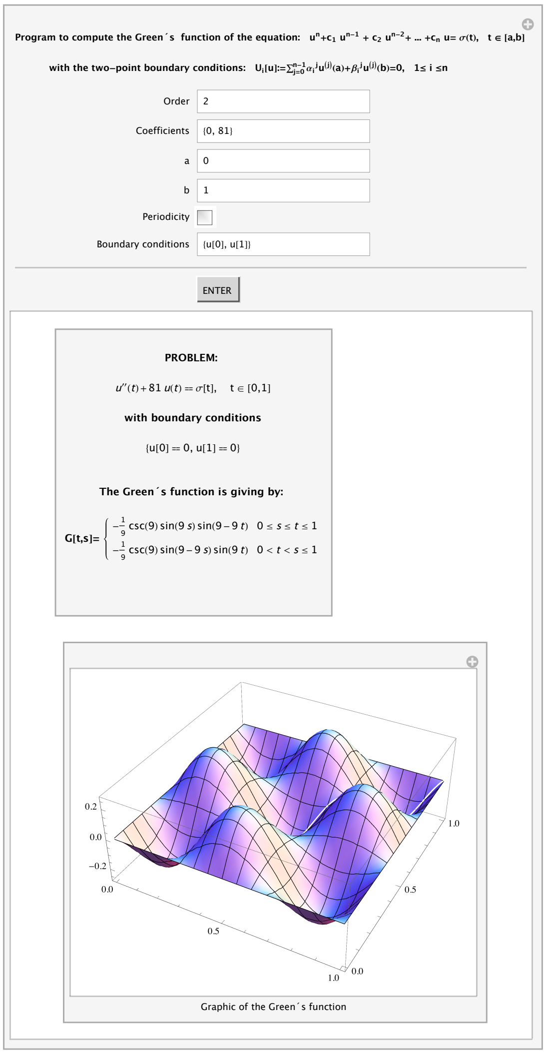

Figure 2: Environment for the calculation of the Green's function

We have also posted a demonstration at the Wolfram site, see [10], where a restricted version of the program can be used without *Mathematica*. In particular, the expression of the Green's function of the following problems can be obtained by downloading the source code of the demonstration:

- $u'(t) \pm m\,u(t) = \sigma(t), \qquad t \in [a,b]; \qquad u(a) = 0;$

- $u'(t) \pm m\, u(t) = \sigma(t), \qquad t \in [a,b]; \qquad u(b) = 0;$
- $u'(t) \pm m\, u(t) = \sigma(t), \qquad t \in [a,b]; \qquad u(a) = u(b);$
- $u''(t) \pm m^2\, u(t) = \sigma(t), \qquad t \in [a,b]; \qquad u(a) = u(b) = 0;$
- $u''(t) \pm m^2\, u(t) = \sigma(t), \qquad t \in [a,b]; \qquad u'(a) = u'(b) = 0;$
- $u''(t) \pm m^2\, u(t) = \sigma(t), \qquad t \in [a,b]; \qquad u^{(i)}(a) = u^{(i)}(b);\ i = 0,\ 1;$
- $u'''(t) \pm m^3\, u(t) = \sigma(t), \qquad t \in [a,b]; \qquad u^{(i)}(a) = u^{(i)}(b);\ i = 0,\ 1,\ 2;$
- $u^{(4)}(t) \pm m^4\, u(t) = \sigma(t), \quad t \in [a,b]; \qquad u(a) = u(b) = 0;\ u'(a) = u'(b) = 0;$
- $u^{(4)}(t) \pm m^4\, u(t) = \sigma(t), \quad t \in [a,b]; \qquad u(a) = u(b) = 0;\ u''(a) = u''(b) = 0;$
- $u^{(4)}(t) \pm m^4\, u(t) = \sigma(t), \quad t \in [a,b]; \qquad u^{(i)}(a) = u^{(i)}(b);\ i = 0,\ 1,\ 2,\ 3.$

Moreover the graph of the corresponding Green's function is plotted for $a = 0$, $b = 1$ and $m \in [0.1, 10]$.

References

[1] Abell, M. L., Braselton, J.P.: Differential equations with *Mathematica*. Elsevier, (2004)

[2] Agarwal, R, P., O'Regan, D.: Ordinary and partial differential equations. With special functions, Fourier series, and boundary value problems. Universitext, Springer, New York, (2009)

[3] Barteneva, I. V., Cabada, A., Ignatyev, A. O.: Maximum and anti-maximum principles for the general operator of second order with variable coefficients. Appl. Math. Comput. **134**, 173-184 (2003)

[4] Bernfeld, S. R., Lakshmikantham, V.: An introduction to nonlinear boundary value problems. Math. Sci. Engr. 109, Academic Press, New York, (1974)

[5] Birkhoff, G., Rota, G. C.: Ordinary differential equations. Fourth edition, John Wiley & Sons, Inc., New York, (1989)

[6] Cabada, A.: The method of lower and upper solutions for second, third, fourth, and higher order boundary value problems. J. Math. Anal. Appl. **185**, 302–320 (1994)

[7] Cabada, A.: The method of lower and upper solutions for third – order periodic boundary value problems. J. Math. Anal. Appl. **195**, 568–589 (1995)

[8] Cabada, A.: Maximum principles for third-order initial and terminal value problems. In:Differential & difference equations and applications, 247–255, Hindawi Publ. Corp., New York, (2006)

[9] Cabada, A., Cid, J. A.: On the sign of the Green's function associated to Hill's equation with an indefinite potential. Appl. Math. Comput. **205**, 303–308 (2008)

[10] Cabada A., Cid, J. A., Máquez-Villamarín, B.: Green's Function, available at `http://demonstrations.wolfram.com/GreensFunction/`, Wolfram Demonstrations Project. Published: October 3, (2011)

[11] Cabada, A., Cid, J. A., Sanchez, L.: Positivity and lower and upper solutions for fourth order boundary value problems. Nonlinear Anal. **67**, 1599–1612 (2007)

[12] Cid, J. A., Franco, D., Minhós, F., Positive fixed points and fourth-order equations. Bull. Lond. Math. Soc. **41**, 72 – 78 (2009)

[13] Coddington, E. A., Levinson, N.: Theory of ordinary differential equations. McGraw-Hill Book Company, Inc., New York-Toronto-London, (1955)

[14] De Coster, C., Habets, P.: Two-point boundary value problems: lower and upper solutions. Mathematics in Science and Engineering, 205. Elsevier B. V., Amsterdam, (2006)

[15] Davies, B.: Integral transforms and their applications. Third edition. Texts in Applied Mathematics, 41. Springer-Verlag, New York, (2002)

[16] Dieudonné, J.: Foundations of Modern Analysis. Pure and Applied Mathematics, Hesperides Press, (2008)

[17] Duffy, D. G.: Green's functions with applications. Studies in Advanced Mathematics. Chapman & Hall/CRC, Boca Raton, FL, (2001)

[18] Fried, H. M.: Green's functions and ordered exponentials. Cambridge University Press, Cambridge, (2002)

[19] Friedman, B.: Principles and techniques of applied mathematics. Reprint of the 1956 original, Dover Publications, Inc., New York, (1990)

[20] Gray, A., Mezzino, M., Pinsky, M. A.: Introduction to ordinary differential equations with *Mathematica*. Springer-Verlag, New York, (1997)

[21] Hakl, R.,Torres, P.: Maximum and antimaximum principles for a second order differential operator with variable coefficients of indefinite sign. Appl. Math. Comput. **217**, 7599-7611 (2011)

[22] Heikkilä, S., Lakshmikantham, V., Monotone iterative techniques for discontinuous nonlinear differential equations. Monographs and Textbooks in Pure and Applied Mathematics, 181, Marcel Dekker, Inc., New York, (1994)

[23] Kythe, P. K., Puri, P., Schäferkotter, M. R.: Partial differential equations and boundary value problems with *Mathematica*. Second revised ed, Boca Raton, Chapman and Hall/CRC, (2003)

[24] Ladde G. S., Lakshmikantham, V., Vatsala, A.S., Monotone iterative techniques for nonlinear differential equations. Pitman, Boston M.A., (1985)

[25] Lynch, S.: Dynamical systems with applications using *Mathematica*. Boston, Birkhäuser, (2007)

[26] Mawhin, J.: Maximum Principle for Bounded Solutions of the Telegraph Equation: The Case of High Dimensions. Progress in Nonlinear Differential Equations and Their Applications, **63**, 343-351 (2005)

[27] Maynard, C. W., Scott, M. R.: Some relationships between Green's functions and invariant imbedding. J. Optimization Theory and Applications **12**, 6-15 (1973)

[28] Novo, S., Obaya, R., Rojo, J.: Equations and Differential Systems (in Spanish). McGraw-Hill, (1995)

[29] Omari, P., Trombetta, M.: Remarks on the lower and upper solutions method for second and third-order periodic boundary value problems. Appl. Math. Comput. **50**, 1-21 (1992)

[30] Peterson, A.: On the Sign of Green's Functions. J. Differential Equations **21**, 167-178 (1976)

[31] Protter M.H., Weinberger, H.F.: Maximum principles in differential equations. Prentice-Hall, (1967)

[32] Roach, G. F.: Green's functions. Second edition, Cambridge University Press, Cambridge-New York, (1982)

[33] Scott, M.: Invariant Imbedding and the Calculation of Internal Values. J. Math. Anal. Appl. **28**, 112–119 (1969)

[34] Stakgold, I.: Green's functions and boundary value problems. Second edition, Pure and Applied Mathematics, John Wiley & Sons, Inc., New York, (1998)

[35] Struwe, M.: Variational methods. Applications to nonlinear partial differential equations and Hamiltonian systems. Springer–Verlag, (2000)

[36] Torres, P. J.: Existence of one-signed periodic solutions of some second-order differential equations via a Krasnoselskii fixed point theorem. J. Differential Equations **190** 643–662 (2003)

[37] Webb, J. R. L. Solutions of nonlinear equations in cones and positive linear operators. J. Lond. Math. Soc. (2) **82** 420 – 436 (2010)

[38] Webb, J. R. L., Infante, G., Franco, D., Positive solutions of nonlinear fourth-order boundary-value problems with local and non-local boundary conditions. Proc. Roy. Soc. Edinburgh Sect. A **138** 427 – 446 (2008)

[39] Zhang, M.: Optimal Conditions for Maximum and Anti-Maximum Principles of the Periodic Solution Problem. Bound. Value Probl. Article ID 410986, 26 pages, (2010)

[40] Zhang, M., Li, W. A.: Lyapunov-type stability criterion using L^α norms. Proc. Amer. Math. Soc. **130**, 3325–3333 (2002)

6 Appendix: *Mathematica* code

The complete code of the program can be downloaded from the site `http://webspersoais.usc.es/persoais/alberto.cabada/index.html`. Recall that *Mathematica* is required on your computer to run the program (we haved used version 8.0.1.0). Alternatively, you can see [10] for a restricted version of the program which works without *Mathematica*.

```
Clear["Global`*"];
Manipulate[
 Module[{d, e, coef, aux, i, j, k, alfa, beta, U, yk, ecrnt, cadenatexto,
   cadenatexto2, rango, condcont2, ecinicial, determinante, soluci}, Off[];
  If[(c ∈ Reals ∧ extb ∈ Reals ∧ exta ∈ Reals) === True,
   Graphic = True, Graphic = False];
  If[(n ∈ Integers && n > 0) === True,
   If[Periodicity == False,
    If[(extb ∈ Reals ∧ exta ∈ Reals ∧ exta ≥ extb ) === True,
     MessageDialog["a must be less than b"]; ecu = 0; l = 0,
     If[n == Length[c] && n == Length[cc],
      l = 0; aux[u_][exta_, extb_] := cc;
      condcont = Table[aux[u][exta, extb][[i]] == 0, {i, 1, Length[c]}];
      Do[alfa[i, j] = Coefficient[aux[u][exta, extb][[i]], u^(j)[exta]],
       {j, 0, Length[c] - 1}, {i, 1, Length[c]}];
      Do[beta[i, j] = Coefficient[aux[u][exta, extb][[i]], u^(j)[extb]],
       {j, 0, Length[c] - 1}, {i, 1, Length[c]}];
      rango = MatrixRank[Join[Table[alfa[i, j], {i, 1, n}, {j, 0, n - 1}],
         Table[beta[i, j], {i, 1, n}, {j, 0, n - 1}], 2]];
      Do[U_i[u_][exta_, extb_] = Sum_{j=0}^{Length[c]-1} (alfa[i, j] * u^(j)[exta] +
           beta[i, j] * u^(j)[extb]), {i, 1, Length[c]}];
      condcont2 = Table[U_i[u][exta, extb] == 0, {i, 1, Length[c]}]; contador = 0;
      If[condcont === condcont2, Label[volver]; contador = contador + 1;
       ecinicial = DSolve[Join[{y^(Length[c])[t] + Sum_{i=1}^{Length[c]} c[[i]] y^(Length[c]-i)[t] == 0},
          Table[y^(i)[0] == 0, {i, 0, Length[c] - 2}], {y^(Length[c]-1)[0] == 1}],
         y, t]; cadenatexto = ToString[ecinicial];
       If[StringMatchQ[cadenatexto, "*Root*"] == False,
        Style[Column[{Row[{Panel[
              Column[{Style[" ", Bold], Style["PROBLEM: ", Bold], Style[" ", Bold],
                Row[{problema[t_] = TraditionalForm[u^(Length[c])[t] +
                      Sum_{i=1}^{Length[c]} c[[i]] u^(Length[c]-i)[t]]; problema[t] == σ[t],
                  ",    t ∈ [", exta, ",", extb, "]"}], Style[" ", Bold],
                Style["with boundary conditions", Bold], Style[" ", Bold],
                condcont2, r[t_] = ComplexExpand[Re[y[t] /. ecinicial[[1]]]];,
                Style[" ", Bold], Style["The Green´s function is given by: ",
                 Bold],
                If[(c ∈ Reals ∧ extb ∈ Reals ∧ exta ∈ Reals) === True,
                 Do[soluci[k] =
                   DSolve[Join[{y^(Length[c])[t] + Sum_{i=1}^{Length[c]} c[[i]] y^(Length[c]-i)[t] == 0},
```

```
            Table[y^(i)[0] == 0, {i, 0, k - 2}], {y^(k-1)[0] == 1},
            Table[y^(i)[0] == 0, {i, k, Length[c] - 1}]], y, t];
          yk[k][t_] = FullSimplify[ComplexExpand[y[t] /.
              Extract[soluci[k], {1, 1}]]];, {k, 1, Length[c]}],
         Do[yk[k][t_] = r^(Length[c]-k)[t] + ∑_{j=k}^{Length[c]-1} c[[Length[c] - j]] r^(j-k)[t];,
          {k, 1, Length[c]}]]; determinante = Chop[
          1. Det[Table[Uᵢ[yk[j]][exta, extb], {i, 1, n}, {j, 1, n}]]];
        If[rango == n,
         If[(determinante == 0 &&
              (c ∈ Reals ⋀ extb ∈ Reals ⋀ exta ∈ Reals)) === True,
          MessageDialog["There is not Green's function"]; Graphic =
           False; l = 0; G[t_, s_] = "There is not unique solution"; ecu = 0;
          aa = exta; bb = extb;, ecuacion := Solve[Table[0 == ∑_{j=0}^{Length[c]-1} beta[
                 i, j] * r^(j)[extb - s] + ∑_{j=1}^{Length[c]} dⱼ[s] Uᵢ[yk[j]][exta, extb] ,
              {i, 1, Length[c]}], Table[dᵢ[s], {i, 1, Length[c]}]];
          If[ecuacion == {},
           MessageDialog["There is not Green's function"]; Graphic = False;
           l = 0; G[t_, s_] = "There is not unique solution";
           ecu = 0; aa = exta; bb = extb;,
           ecu = 1; coef := Sort[Extract[ecuacion, {1}]];
           Do[e[i][s_] := dᵢ[s] /. Extract[coef, {i}], {i, 1, n}] ;
           h[t_, s_] := ∑_{i=1}^{Length[c]} Simplify[e[i][s]] yk[i][t];
           G1[t_, s_] = Simplify[Chop[r[t - s] + h[t, s]]];
           G2[t_, s_] = Simplify[Chop[h[t, s]]];
           G[t_, s_] = { G1[t, s]  exta ≤ s ≤ t ≤ extb
                       { G2[t, s]  exta < t < s ≤ extb ;
           aa = exta; bb = extb; ]],
         MessageDialog["The boundary condicions are linearly dependent"];
         Graphic = False; l = 0; G[t_, s_] = "There is not unique solution";
         ecu = 0; aa = exta; bb = extb;]
        , Row[{Style["G[t,s]= ", Bold], TraditionalForm[G[t, s]]}],
        Style[" ", Bold], Style[" ", Bold], Style[" ", Bold]}, Center]],
     Style[" ", Bold], If[Graphic == False, , Manipulate[If[ecu == 1, Plot3D[
         G[t, s], {s, aa, bb}, {t, aa, bb}], "Cannot show the graphic"],
        FrameLabel → "Graphic of the Green´s function"]]}]}]]],
  c = N[c]; If[contador == 2, MessageDialog[
     "The program cannot calculate the Green´s function"];
    l = 0; ecu = 0; aa = exta; bb = extb; , Goto[volver]]],
  MessageDialog["The boundary conditions are not valid"];
  l = 0; ecu = 0; aa = exta; bb = extb; ],
 MessageDialog["Vector of coefficients or Boundary conditions:
```

```
        LENGTH INCORRECT"]; l = 0; ecu = 0; aa = exta; bb = extb; ]],

    If[(extb ∈ Reals ∧ exta ∈ Reals ∧ exta ≥ extb ) === True,
     MessageDialog["a must be less than b"]; ecu = 0; l = 0,
     contador2 = 0;
     If[n == Length[c],
      Off[DSolve::bvnul]; l = 0; Label[volver2];
      ecrnt = DSolve[Join[{y^(Length[c])[t] + Sum[c[[i]] y^(Length[c]-i)[t], {i, 1, Length[c]}] == 0},
          Table[y^(i)[exta] - y^(i)[extb] == 0, {i, 0, Length[c] - 2}],
          {y^(Length[c]-1)[exta] - y^(Length[c]-1)[extb] == 1}], y, t];
      cadenatexto2 = ToString[ecrnt]; contador2 = contador2 + 1;
      If[StringMatchQ[cadenatexto2, "*Root*"] == False,
       Style[Column[{Row[{Panel[Column[{Style[" ", Bold], Style["PROBLEM: ",
                Bold], Style[" ", Bold], Row[{problema[t_] = TraditionalForm[
                   u^(Length[c])[t] + Sum[c[[i]] u^(Length[c]-i)[t], {i, 1, Length[c]}]]; problema[t] == σ[t],
                 ",    t ∈ [", exta, ",", extb, "]"}], Style[" ", Bold],
               Style["with boundary conditions", Bold], Style[" ", Bold],
               condcont = Table[u^(i)[exta] - u^(i)[extb] == 0, {i, 0, Length[c] - 1}];
               condcont, Style[" ", Bold], Style[
                "The Green´s function is given by: ", Bold],
               If[ecrnt == {},
                 MessageDialog["There is not Green's function"];
                 Graphic = False; G[t_, s_] = "There is not unique solution";
                 ecu = 0; aa = exta; bb = extb;,
                 ecu = 1; r[t_] = ComplexExpand[Re[y[t] /. ecrnt[[1]]]];
                 ecu = 1; G1[t_, s_] = Simplify[Chop[r[exta + t - s]]];
                 G2[t_, s_] = Simplify[Chop[r[extb + t - s]]];
                 G[t_, s_] = { G1[t, s]  exta ≤ s ≤ t ≤ extb
                               G2[t, s]  exta < t < s ≤ extb ; aa = exta; bb = extb;];
               , Row[{Style["G[t,s]= ", Bold], TraditionalForm[G[t, s]]}],
               Style[" ", Bold], Style[" ", Bold], Style[
                " ", Bold]}, Center]], Style[" ", Bold],
            If[Graphic == False, , Manipulate[If[ecu == 1, Plot3D[G[t, s],
                 {s, aa, bb}, {t, aa, bb}], "Cannot show the graphic"], FrameLabel →
               "Graphic of the Green´s function"]]}]}]], c = N[c]; If[contador2 == 2,
        MessageDialog["The program cannot calculate the Green´s function"];
        l = 0; ecu = 0; aa = exta; bb = extb; , Goto[volver2]]],
      MessageDialog["Length of coefficients´ vector or boundary condictions
         INCORRECT"]; l = 0; ecu = 0; aa = exta; bb = extb; ]]],
   MessageDialog["Order must be a positive integer"]; l = 0; ecu = 0;
   aa = exta; bb = extb;]],
Style["Program to compute the Green´s  function of the equation:
   u^n+c_1 u^(n-1) + c_2 u^(n-2)+ ... +c_n u= σ(t),   t ∈ [a,b]", Bold],
Style[Column[{"                                              "}, Center], Bold],
```

```
Style["           with the two-point boundary

   conditions:   U_i[u]:= Sum_{j=0}^{n-1} α_i^j u^(j)(a) + β_i^j u^(j)(b) = 0,    1≤ i ≤n", Bold],
Style["                                        ", Bold],
{{n, 2, "                                                         Order"}},
{{c, {0, 0}, "Coefficients"}},
{{exta, 0, "a"}}, {{extb, 1, "b"}},
{{Periodicity, False, "Periodic conditions"}, {False, True}, ControlType → Checkbox},
{{cc , If[Periodicity == True, , {u[exta], u[extb]}], "Boundary conditions"},
 FieldHint → "Periodic", Enabled → Not[Periodicity == True]},
Delimiter, {{l, , " " }, {ENTER}, ControlType → Setter},
ControlPlacement → Top, TrackedSymbols -> {l}, Alignment → Center]
```