Green's Functions with Reflection

User's manual

Alberto Cabada Fernández (USC)

José Ángel Cid Araújo (UVIGO)

Fernando Adrián Fernández Tojo (USC)

Beatriz Máquez Villamarín (USC)

Universidade de Santiago de Compostela Universidade de Vigo

1. Language

Code written in Mathematica 8.0.1.0

Version: October 2014.

2. Environment

Notebook Mathematica.

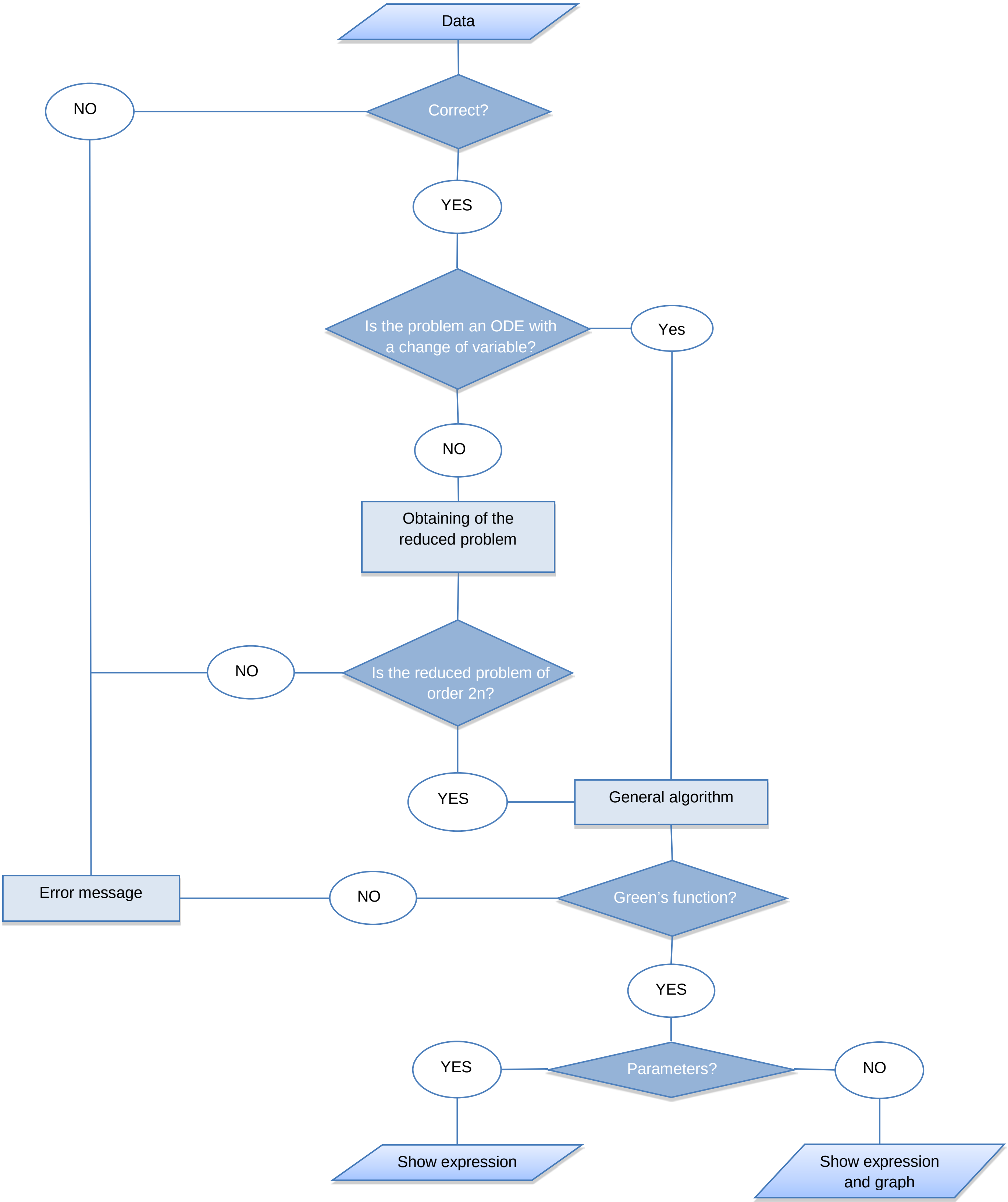
3. Name of the file

GFR.nb

4. Abstract

This Mathematica package provides a tool valid for calculating the explicit expression of the Green's function related to an n^{th} – order linear differential equation with reflection and constant coefficients, coupled with two – point linear boundary conditions.

5. Flowchart



Data
Correct?
NO
YES
Is the problem an ODE with a change of variable?
Yes
NO
Obtaining of the reduced problem
Is the reduced problem of order 2n?
NO
YES
General algorithm
Error message
NO
Green's function?
YES
Parameters?
YES
NO
Show expression
Show expression and graph

6. User's manual

The program *Green's Functions with Reflection* calculates the Green's function, $G(t,s)$, of the boundary value problem given by a linear n^{th} - order differential equation with reflection and constant coefficients

$$\sum_{j=0}^{n} a_j u^{(j)}(-t) + \sum_{j=0}^{n} b_j u^{(j)}(t) = \sigma(t), \qquad t \in [a,b], \quad (1)$$

together with the boundary conditions

$$\sum_{j=0}^{n-1} \alpha_i^j u^{(j)}(a) + \beta_i^j u^{(j)}(b) = 0, \qquad 1 \le i \le n. \qquad (2)$$

Definition. A Green's function of problem (1)-(2) is any function $G(t,s)$ such that, for any $\sigma \in L^1([a,b])$,

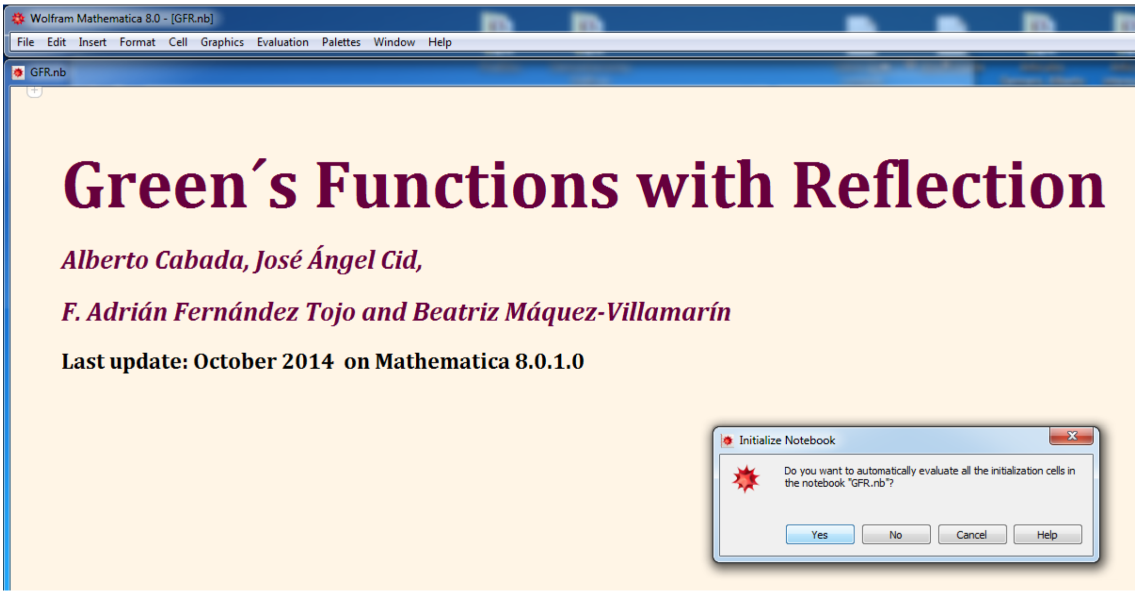
$$u(t) = \int_a^b G(t,s)\sigma(s)ds$$

is the unique solution of problem (1)-(2).

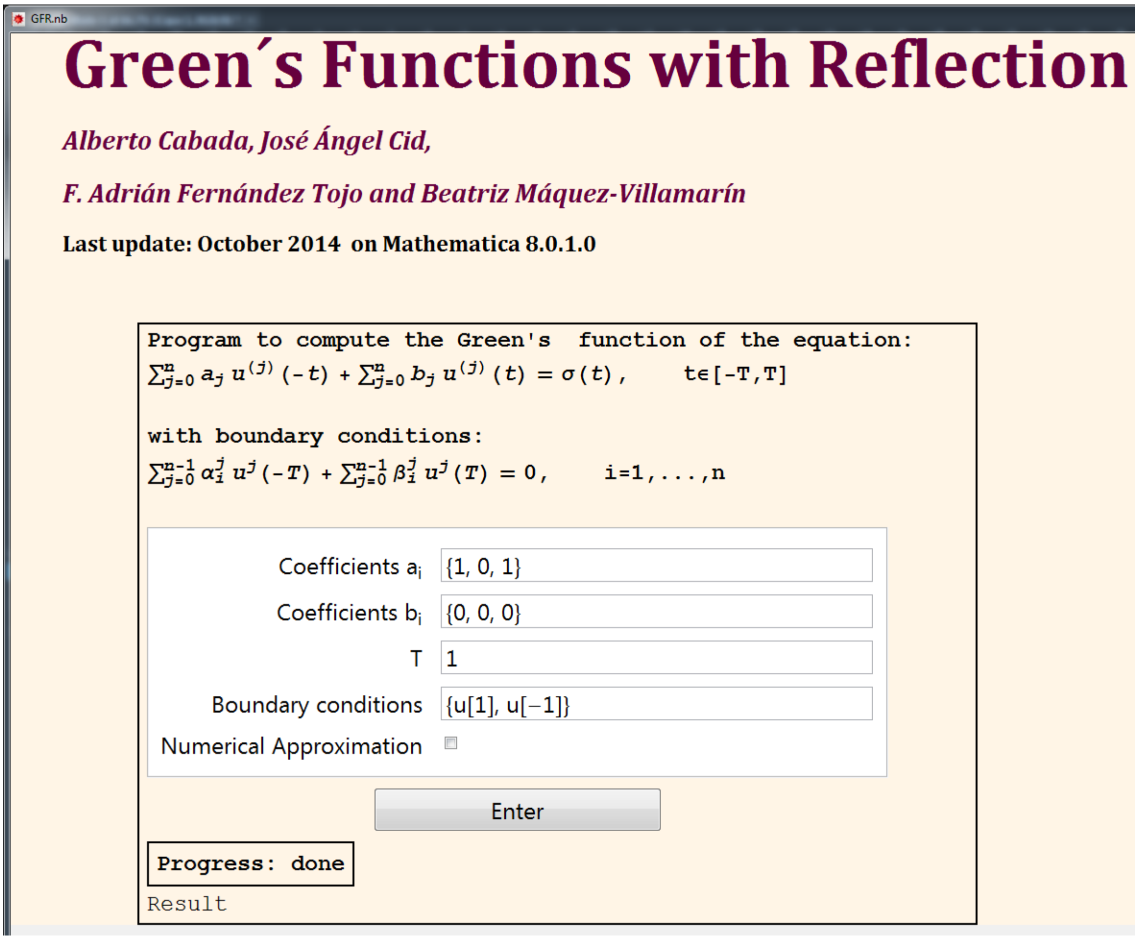
6.1. The *Mathematica* notebook

The *Green's Functions with Reflection* is a *Mathematica* notebook with a dynamic environment. In order to run the program *Wolfram Mathematica* is needed on the user's computer. This notebook is intended for version 8.0.1.0 but it also works on less recent versions.

After opening the file, a message will appear. The user must press yes in order to authorize Mathematica to initialize the program.



After this, the input cells of the notebook will appear, already filled with an example.



If the user presses “Enter” the output cells will appear revealing the results of the computation.

Enter

Progress: done

PROBLEM:

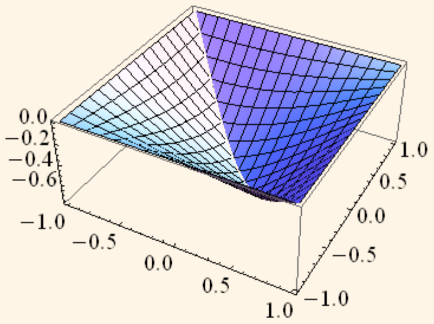
$$u''(-t) + u(-t) == \sigma[t], \qquad t \in [-1,1]$$

with boundary conditions

$$\{u[1] == 0,\ u[-1] == 0\}$$

The Green´s function is giving by:

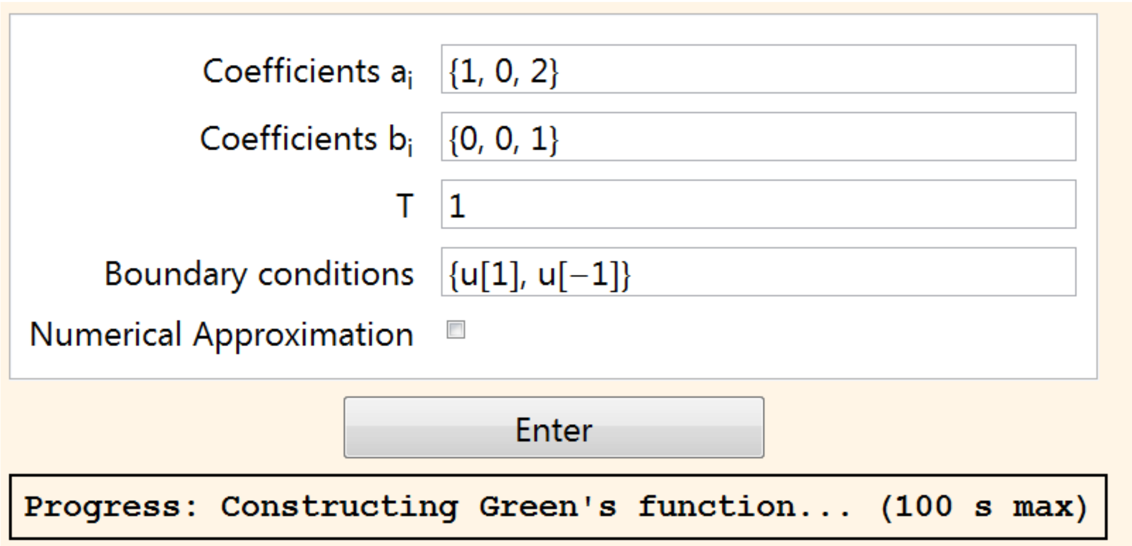
$$G[t,s] = \begin{cases} -\csc(2)\sin(1-s)\sin(1-t) & -1 \le s \le 1 \wedge -1 \le t \le 1 \wedge s+t \ge 0 \\ -\csc(2)\sin(s+1)\sin(t+1) & -1 \le s \le 1 \wedge -1 \le t \le 1 \wedge s+t < 0 \end{cases}$$



In order to preserve the stability of the program through uses, the input cell is protected against writing and the file cannot be saved.

The program has two different areas: *input* and *output,* both separated by the "Enter" button. The data must be entered in the input area by using the *Mathematica* notation. After that, it must be pushed "Enter" to start the execution and the result will be shown on the *output* area, which is divided again in two areas: *analytical output* and *graphical output*.

While running, the steps of the computation will be shown in the "Progress" frame. These messages will be, in order, "Processing data...", "Solving homogeneous equation...", "Computing fundamental matrix...", "Constructing Green's function... (100 s max)" and "done". Usually, the step that takes the longest is the construction of the Green's function. The "100 s max" comment makes reference to the total time limit set for those Mathematica commands during this process which can be aborted after some time giving a valid result, like, for instance "Simplify" or "FullSimplify". This does not mean that other operations on which no time limit can be placed, cannot make the whole process take longer.

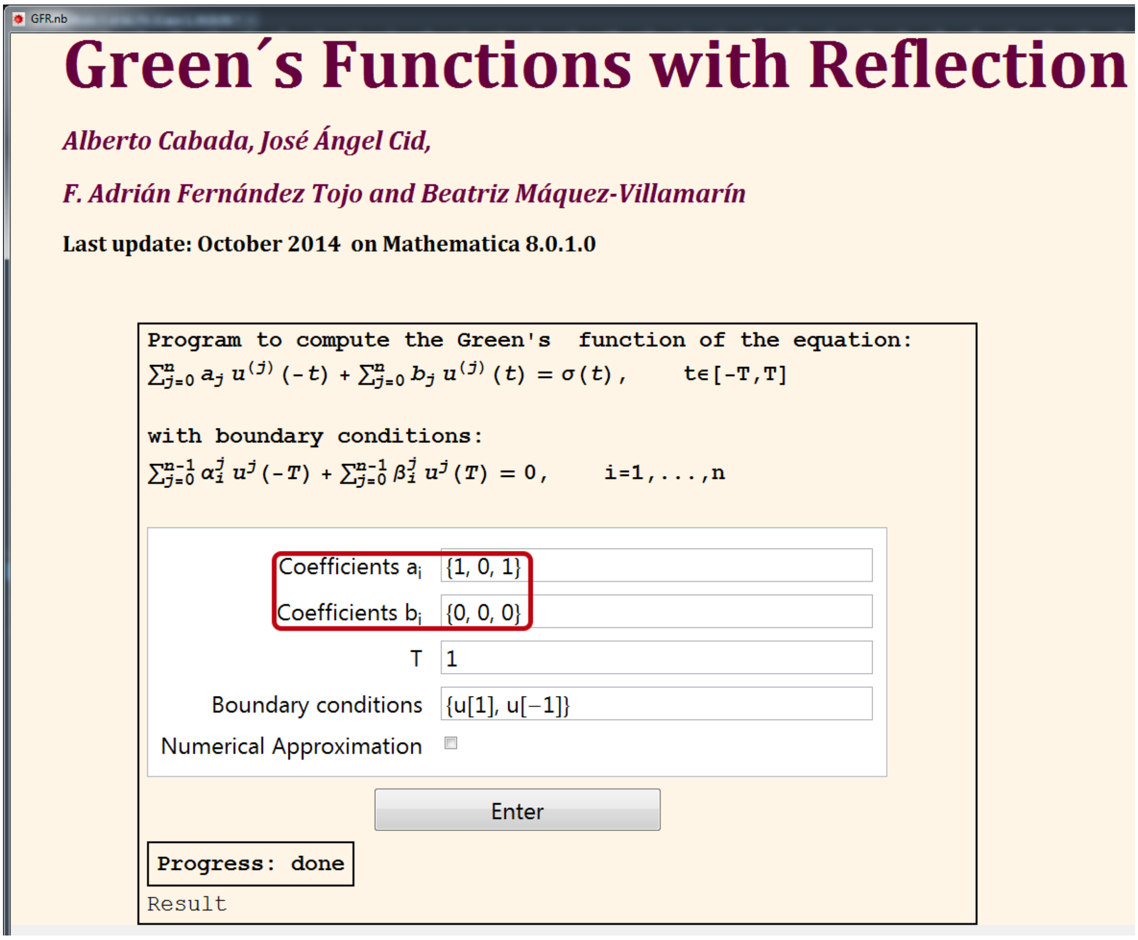


6.2. The data

To start an execution the following boxes must be completed: **Coefficients a_i**, **Coefficients b_i**, **T** and **Boundary conditions.** Notice that all the data have to be entered in *Mathematica* syntax. Some examples are showing in the following table:

	***Mathematica* syntax**	**Examples**
Power	^	m^2
Multiply	* or *empty space*	c*x, c x
Divide	/	1/2, -3/7
Constants	Pi, E (for other constants see the *Mathematica* help page)	2 Pi, E^(-1)
Functions	Sin[x], Cos[x], Tan[x], Log[x], Sqrt[x] (for other functions see the *Mathematica* help page)	2 Sin[3 x], Sqrt[2]
Lists, vectors	{ , , … }	{1,3,2}, {u[0],u'[Pi]}
Derivative	'	u'[0], u''[2], u'''[2*Pi]
Grouping terms	( )	m^(2*x)

The first step consists on introducing the coefficient vectors (both the one related to the terms depending on "-t" and the one related to the terms depending on "t") of the differential equation with the same length as the order of the equation. The order of the coefficients is $\{a_0, a_1, \dots, a_n\}$ and $\{b_0, b_1, \dots, b_n\}$ respectively.



The coefficients of the differential equation must be real constants or symbolic parameters. If those coefficients are exact expressions, (for instance Pi, Sqrt[2], Cos[1/7], 3…) the computations made by *Mathematica* will be exact as the final result.

However if any of the introduced constants is an approximate number (for instance: 1.0, 3.14159, Sqrt[2.0], -0.5,…), the final result will be also approximate. If the "Numerical Approximation" check box is clicked the coefficients, except in the case there are symbolic parameters involved, will be automatically transformed into numerical expressions. This greatly reduces de computation time, and even allows to obtain results when symbolically is not possible, either because of a too large output or because some step in the computation process has exceeded the maximum time allocated for it.

Coefficients a_i	{1., 2.}
Coefficients b_i	{0., 1.}
T	1
Boundary conditions	{u[1]}
Numerical Approximation	☑

Enter

Progress: done

PROBLEM:

$$2.\,u'(-t)+1.\,u'(t)+1.\,u(-t)+0.==\sigma[t],\qquad t\in[-1,1]$$

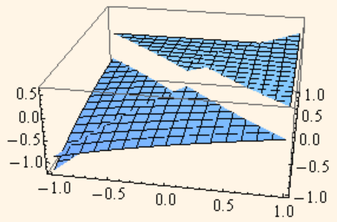
with boundary conditions

{u[1] == 0}

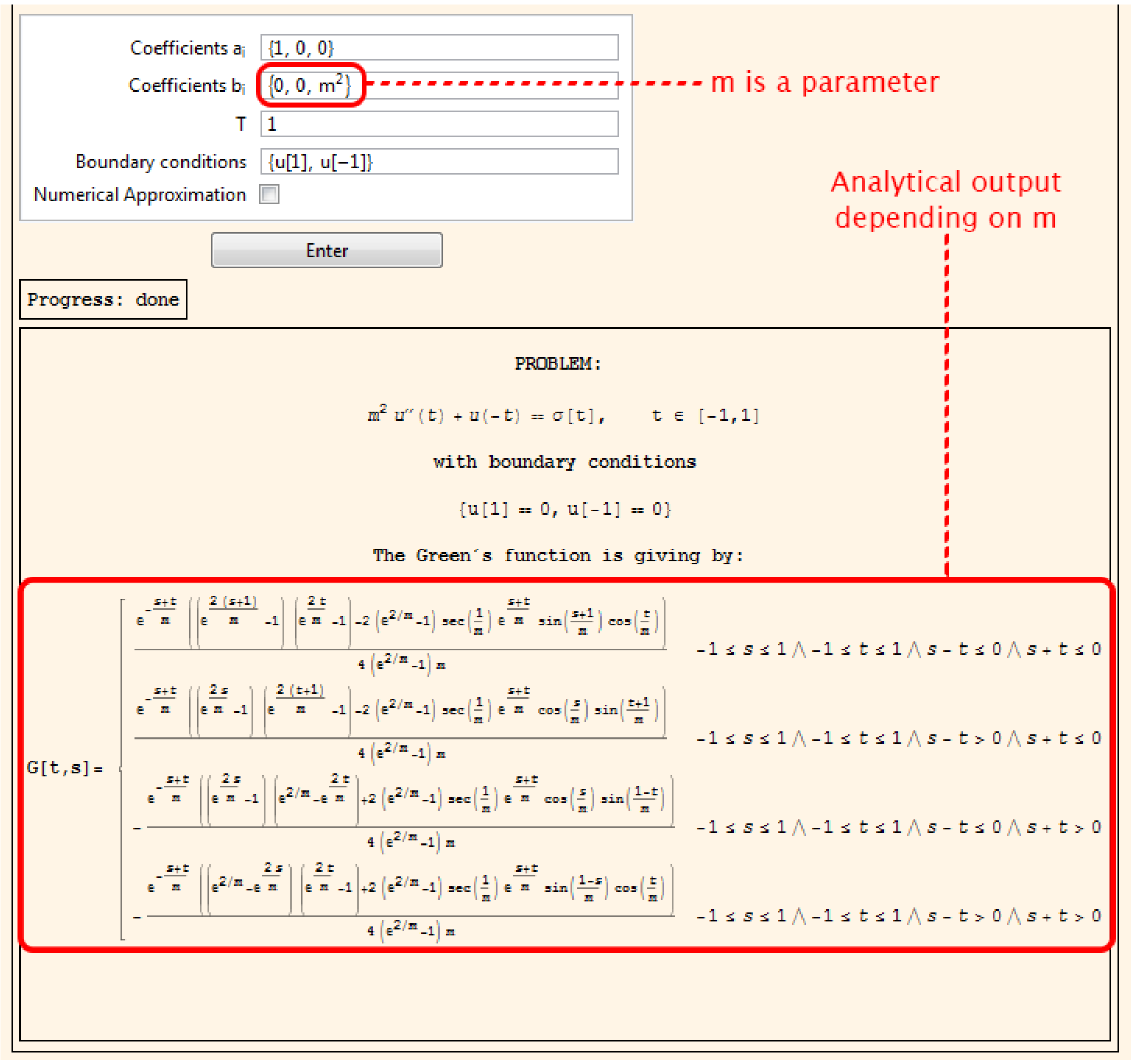
The Green´s function is giving by:

G[t,s]=

$$\begin{cases} -0.33\left(e^{-0.58 s}\left(0.63+0.2 e^{1.2 s}\right)\sinh(0.58 t)+1.9 e^{-0.58 s-0.58 t}+0.5 e^{0.58 s-0.58 t}+0.5 e^{0.58 t-0.58 s}+0.13 e^{0.58 s+0.58 t}+e^{-0.58 s}\left(-0.34 e^{1.2 s}-1.1\right)\cosh(0.58 t)\right) & -1\le s\le 1\land -1\le t\le 1\land s-t\le 0\land s+t\le 0 \\ -0.33\left(e^{-0.58 s}\left(0.63+0.2 e^{1.2 s}\right)\sinh(0.58 t)+1.9 e^{-0.58 s-0.58 t}+0.13 e^{0.58 s+0.58 t}+e^{-0.58 s}\left(-0.34 e^{1.2 s}-1.1\right)\cosh(0.58 t)\right) & -1\le s\le 1\land -1\le t\le 1\land s-t>0\land s+t\le 0 \\ -0.33\left(e^{-0.58 s}\left(0.63+0.2 e^{1.2 s}\right)\sinh(0.58 t)+0.5 e^{0.58 s-0.58 t}+0.5 e^{0.58 t-0.58 s}+e^{-0.58 s}\left(-0.34 e^{1.2 s}-1.1\right)\cosh(0.58 t)\right) & -1\le s\le 1\land -1\le t\le 1\land s-t\le 0\land s+t>0 \\ -0.33\left(e^{-0.58 s}\left(0.63+0.2 e^{1.2 s}\right)\sinh(0.58 t)+e^{-0.58 s}\left(-0.34 e^{1.2 s}-1.1\right)\cosh(0.58 t)\right) & -1\le s\le 1\land -1\le t\le 1\land s-t>0\land s+t>0 \end{cases}$$



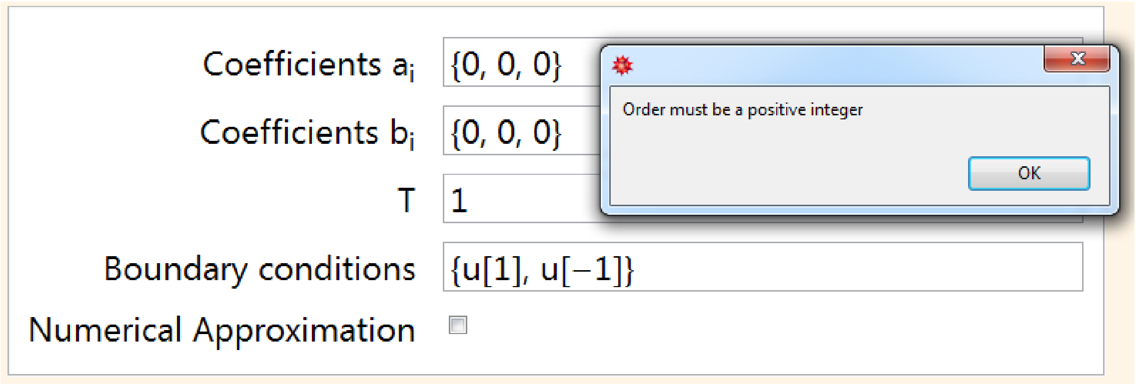
Entering parameters is allowed in the different boxes, but in this case the output will be only analytical and not graphical.



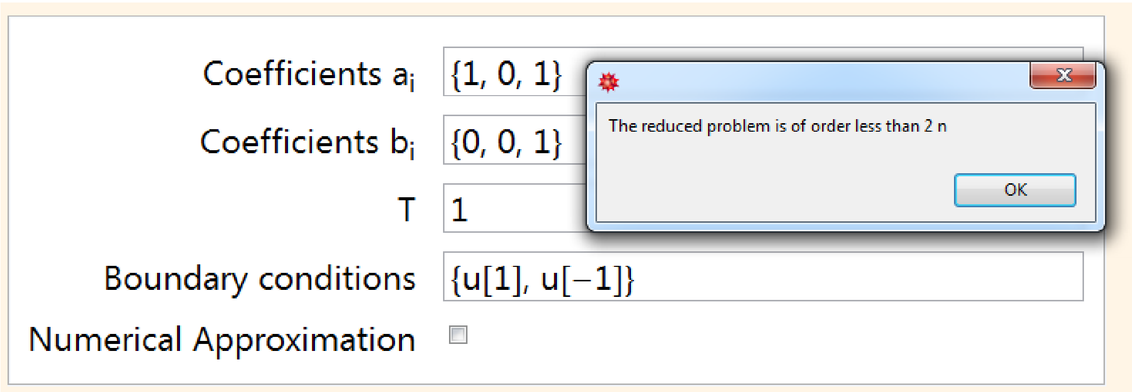
The box **Boundary Conditions** must be a vector of the same length as the order, *n*, of the differential equation. Moreover the boundary conditions must depend linearly on *u* and its derivatives up to the *n-1* order and they must be evaluated at the endpoints of the interval. The vector with the boundary conditions will be matched to zero by the program. For instance, to use the boundary conditions *u(-1)=u(1), u'(-1)= – u'(1)*, the vector {u[-1] – u[1],u'[-1]+u'[1]} must be entered.

6.3. Errors

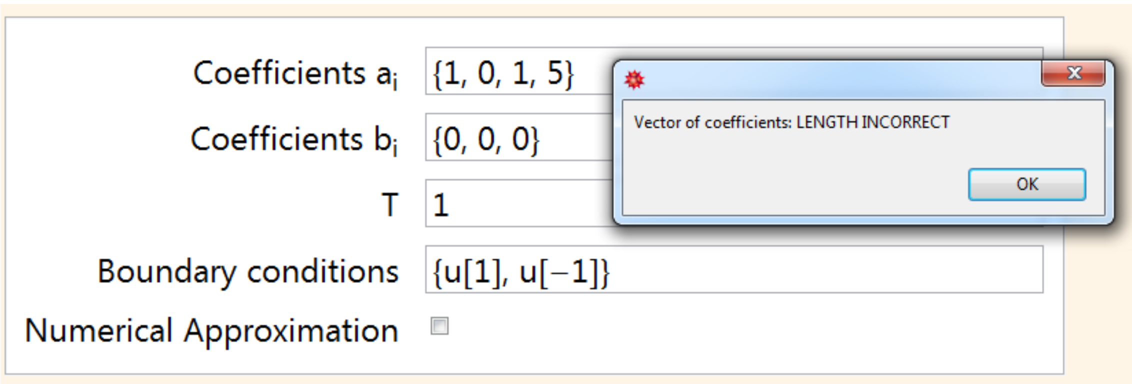
The program will detect when the order of the equation is not correct.



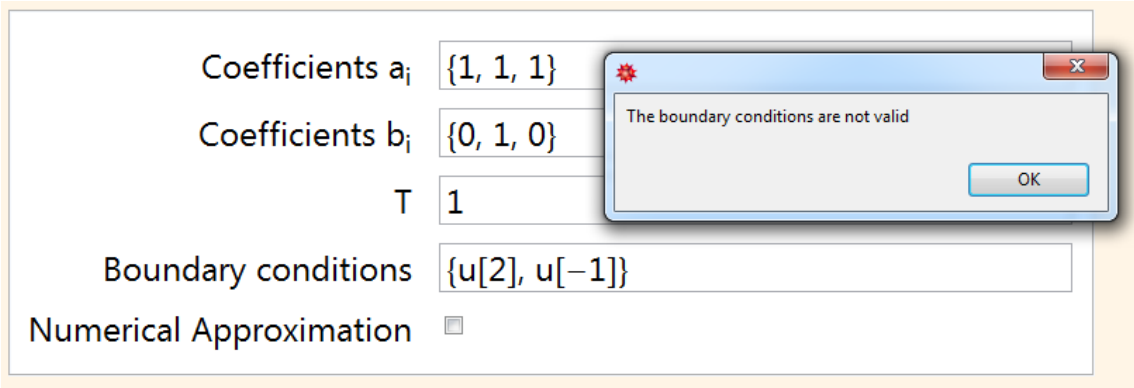
Also, when $a_n = \pm b_n$, the program will not be able to compute a Green's function. This is related to the theoretical aspects of the reduction algorithm. In such cases a Green's function may not exist and a "case-by-case" study must be done.



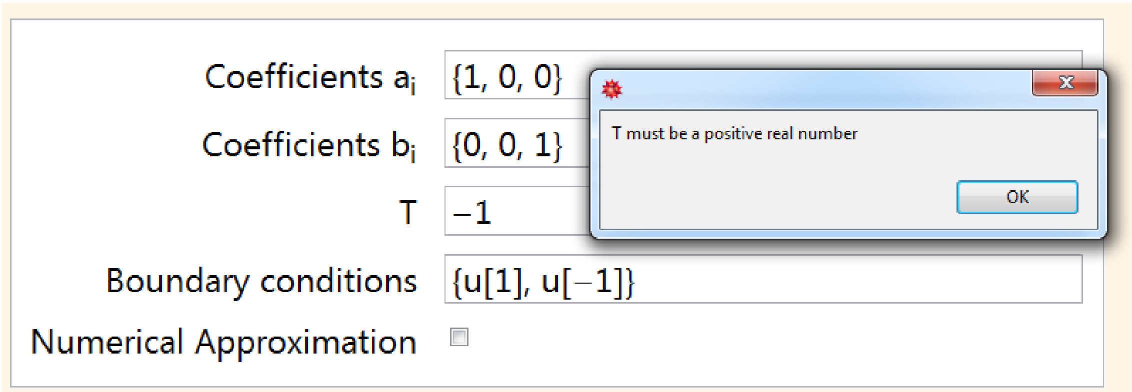
If the number of coefficients a_i is not the same as the number of coefficients b_i the program will warn us.



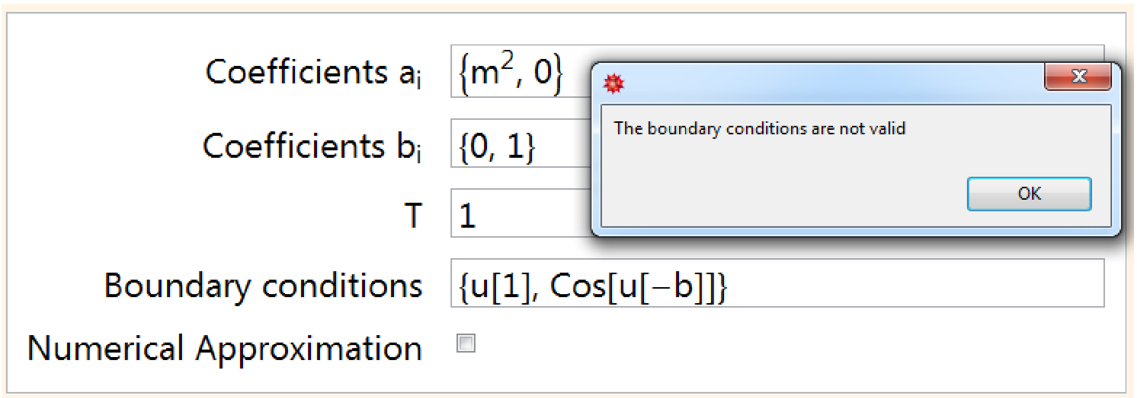
The boundary conditions must be evaluated at the points T and / or –T for the given value of T. Otherwise, an error message will appear.



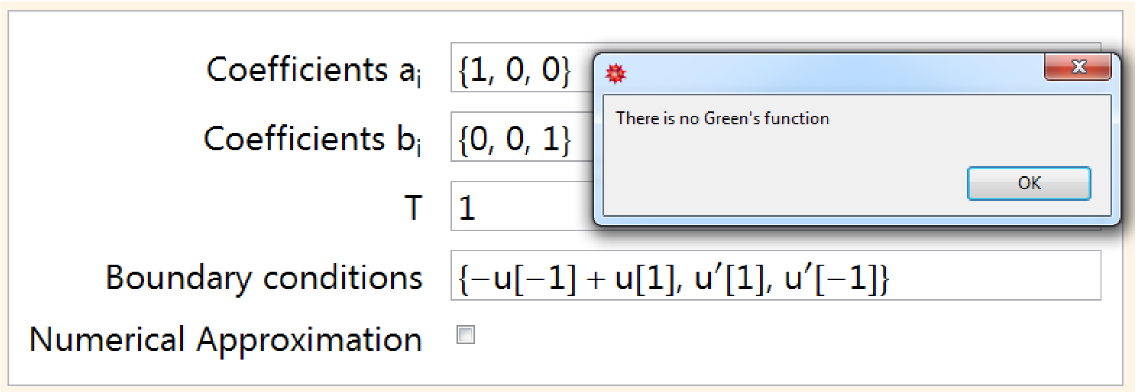
The value of T must be a positive number (or a parameter).



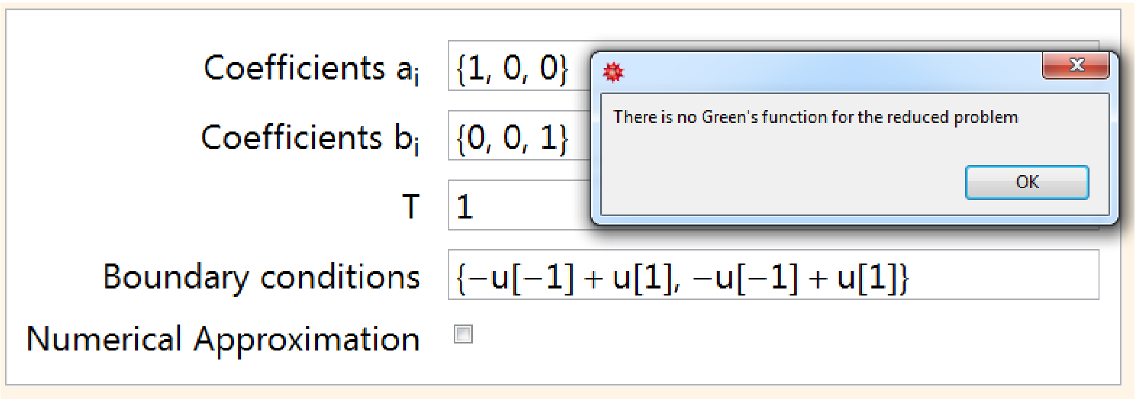
If there is an error or an invalid expression in the input of the boundary conditions the program will warn us.



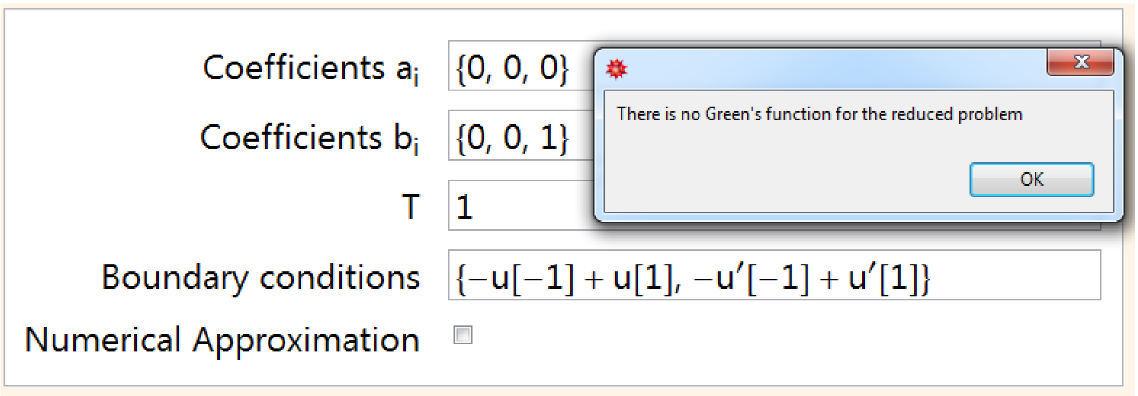
If there is an excessive number of boundary conditions there will not be a Green's function.



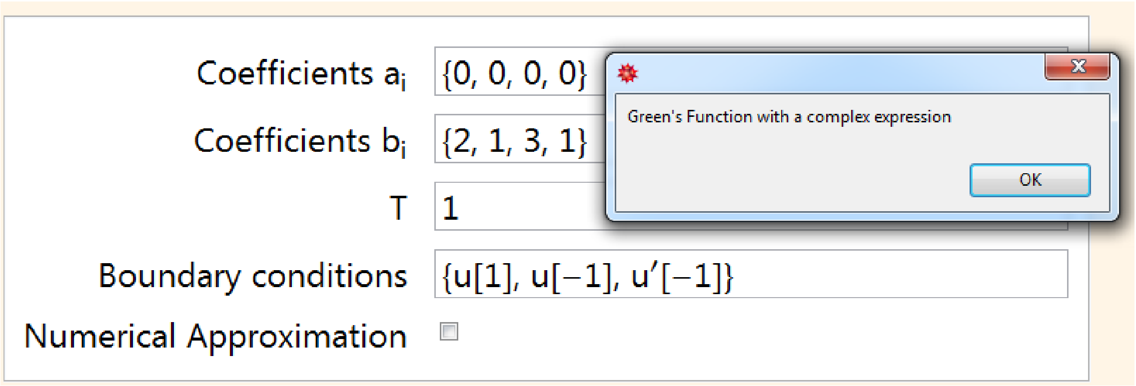
If the number of boundary conditions is insufficient or they are linearly dependant there will not be a Green’s function.



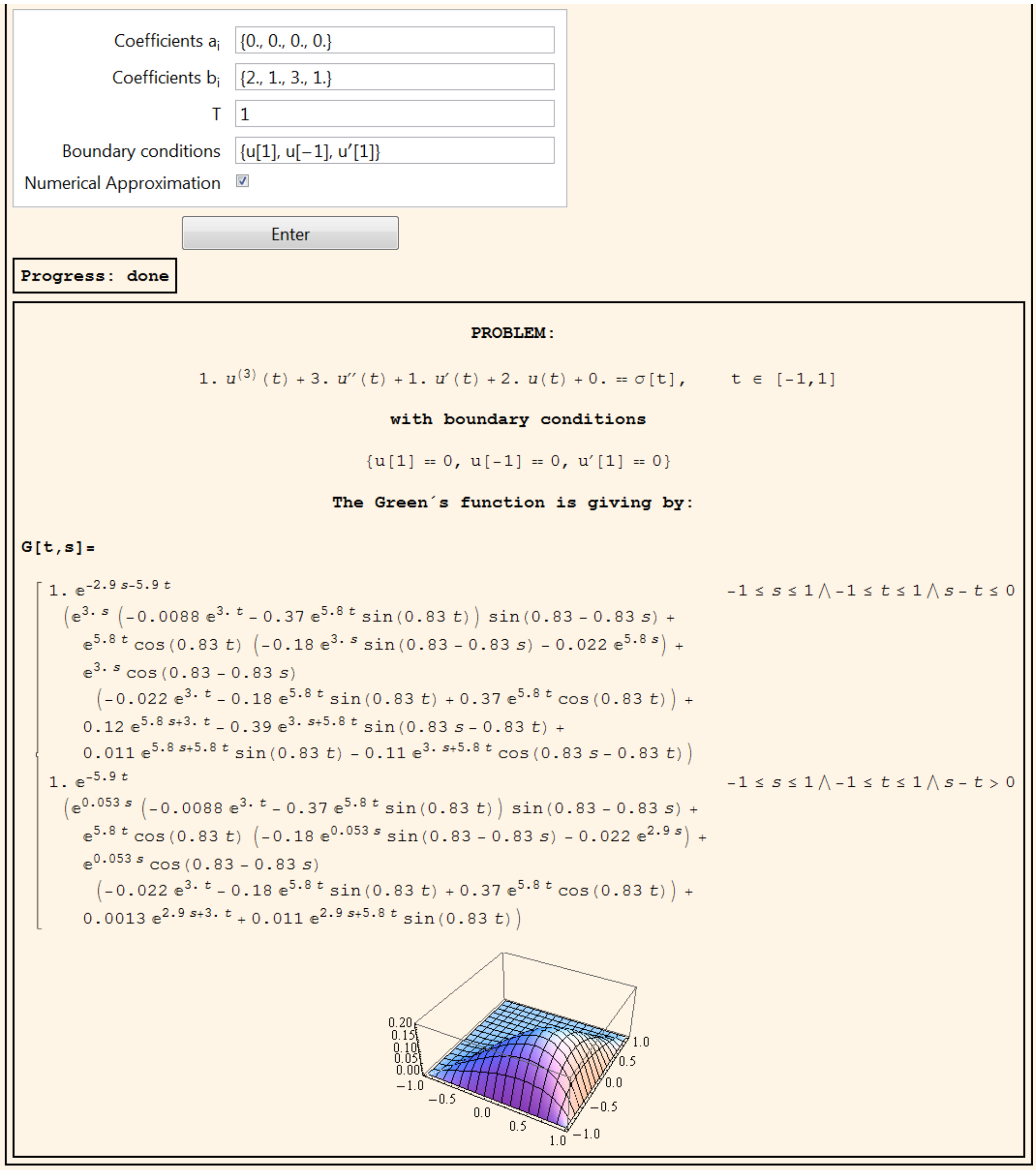
The resonant problems, i. e., when the Green’s function doesn’t exist, are also detected by the program.



In some cases Mathematica will not be able to compute explicitely the roots of the polynomials involved in the computation. In such case the following message will appear.



In this case, Mathematica will be able to compute a numerical solution using the "Numerical Approximation" option.



Some examples have been detected where *Mathematica* was not able to show the expression obtained for the Green's function on the notebook. In this case the program seems blocked. The evaluation can be aborted by using "Evaluation -> Interrupt Evaluation" on the *Mathematica* menu. Open the notebook again and try to solve the problem using the "Numerical Approximation" option.

6.4. Global variables after the execution

The main goal of this program is to obtain the expression of the Green's function in the most standard way. Is for this that some variables of the program are global, so, after an execution, the user can work directly with them on the *Mathematica* notebook. Namely, the Green's function, G[t,s], is a global variable, in consequence if, after an execution, the user writes "G[t,s]" on a new input cell of *Mathematica,* the program gives its expression. In this way it is possible to manipulate or plot it at the convenience of the user.

Bibliography

[Ca1] 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).

[Ca2] Cabada, A., Green's Functions in the Theory of Ordinary Differential Equations. Springer (2014).

[CaCiMa1] Cabada A., Cid, J. A., Máquez-Villamarín, B.: Computation of Green's functions for Boundary Value Problems with *Mathematica,* Applied Mathematics and Computation. 219:4, 1, 1919–1936 (2012).

[CaCiMa2] 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).

[CaTo1] Cabada, A; Tojo F. A. F. Comparison results for first order linear operators with reflection and periodic boundary value conditions. Nonlinear Analysis: Theory, Methods & Applications 78, 32–46 (2013).

[CaTo2] Cabada, A; Tojo F. A. F. Existence results for a linear equation with reflection, non-constant coefficient and periodic boundary conditions. Journal of Mathematical Analysis and Applications 412 (1), 529-546 (2014).

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

[NOR] Novo S., Obaya, R., Rojo, J.: Equations and Differential Systems (in Spanish), McGraw-Hill, (1995)

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