RESEARCH ARTICLE OPEN ACCESS

Solving engineering problems using the finite difference method applied to Scilab

Borges, Jacques Cousteau da Silva*
**(Department of Industry, IFRN, Natal-RN-Brazil email: cousteau.borges@ifrn.edu.br)*

ABSTRACT
This paper describes the procedures adopted, and the results found, in the numerical resolution of a problem of plate flexion, through the finite difference method. The two-dimensional problem was proposed as being a plate that is partially subjected to a uniformly distributed load, and rests on an elastic base, set at its ends and supported on its sides. Given the boundary conditions, an algorithm with the finite difference method was developed. The programming language used was SciLab, and the results were expressed both numerically and graphically. Theoretical fundamentals, program elements and the analysis of their performance are expressed in detail in the following text, including elements that facilitate the use in engineering classes, following a didactic approach, so that it can be reproduced in classrooms of courses of engineering.
Keywords – Finite differences, taylor series, scilab, engineering education

Date of Submission: 20-11-2017 Date of acceptance: 14-12-2017

I. INTRODUCTION

The finite difference method is a numerical alternative to approximate resolution of relatively complex problems, which have no analytical solution, or when it is difficult to solve. In summary, the method transforms the differential equations that govern the phenomenon, into an algebraic system of equivalent differential equations. It is from the Taylor series that the base equations employed are deduced [1][2].

Classically, the Taylor series approximates a function of x, from the successive sum of its derivatives, given two points of the curve, separated by a Δx forwards or backwards. So we have:

$$f(x \pm \Delta x) = \mp f(x) \pm \Delta x \frac{df}{dx} \mp \frac{(\Delta x)^2}{2!}\frac{d^2f}{dx^2} \pm \frac{(\Delta x)^3}{3!}\frac{d^3f}{dx^3}$$

$$f(x \pm \Delta x) = \sum_n \pm(-1)^n \frac{(\Delta x)^n}{n!}\frac{d^nf}{dx^n}$$

The series has infinite terms, being necessary to truncate it in some point, according to the wanted precision. Initially, let's stop at the second term. Regrouping the plots, we can express the approximation of the first derivative as follows:

$$\frac{df}{dx} \approx \frac{f(x) - f(x - \Delta x)}{\Delta x} \quad or \quad \frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

The two expressions are equally valid. The first is known as the descending form (difference to the back), and the second as the ascending form (difference the front). The first form is more used, since, physically, we know the recent past of a function, but the future is always an unknown. Summing up the two expressions, we obtain the "centered" form of the series (or central finite difference), finally expressing the first derivative as:

$$2\frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{\Delta x}$$

If we repeat the procedure, but now truncating the series in the second derivative, we will obtain the difference equation, and replacing what we already know in relation to the central form of the first derivative:

$$\frac{d^2f}{dx^2} \approx \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{(\Delta x)^2}$$

In an analogous way, we can perform this procedure to approximate the derivative of any order, of a given function. To conclude, the approximate third and fourth order expressions can be represented by:

$$\frac{d^3f}{dx^3} \approx \frac{f(x + 2\Delta x) - 2f(x + \Delta x) + 2f(x - \Delta x) - f(x - 2\Delta x)}{2(\Delta x)^3}$$

$$\frac{d^4f}{dx^4} \approx \frac{f(x + 2\Delta x) - 4f(x + \Delta x) + 6f(x) - 4f(x - \Delta x) + f(x - 2\Delta x)}{(\Delta x)^4}$$

With these equations, we can apply the finite difference method to the proposed problem, which is governed by a partial fourth-order differential equation.

II. PROBLEM DEFINITION: BENDING PLATE

A given structuring element can be defined as being one-dimensional, two-dimensional, or three-dimensional, although everything in the physical world is in fact three dimensions. Basically, when the dimensions in a given coordinate is much smaller than the others, it is considered negligible. In the case of plates, the thickness (or height) is much

smaller than the length and width, the plate being a Bidimensional element. From the general theory of plates the following equation is obtained:

$$D\left(\frac{\partial^4 w}{\partial x^4}+2\frac{\partial^4 w}{\partial x^2\partial y^2}+\frac{\partial^4 w}{\partial y^4}\right)+kw=p(x,y)$$

Where: ω represents the transverse displacement of a point on the plate. P is a function of x and y, and gives us the intensity and shape of the charge applied to that plate. k is the elastic constant of the base on which the plate rests. Finally, D is the flexural rigidity of the plate, which is given by the equation: $D=\frac{E\cdot\delta^3}{12(1-\upsilon^2)}$ being E the Young's modulus, δ the plate thickness and υ the Poisson coefficient. With the exception of transverse displacement, all these variables must be provided to solve the problem. Rearranging the equation we have:

$$\left(\frac{\partial^4 w}{\partial x^4}+2\frac{\partial^4 w}{\partial x^2\partial y^2}+\frac{\partial^4 w}{\partial y^4}\right)+\frac{k}{D}w=\frac{p(x,y)}{D}$$

Now each term of the derivatives of w will be expressed by the equivalent center difference equation:

$$\frac{\partial^4 w(x,y)}{\partial x^4}=\frac{w(x+2\Delta x)-4w(x+\Delta x)+6w(x)-4w(x-\Delta x)+w(x-2\Delta x)}{(\Delta x)^4}$$

$$\frac{\partial^4 w(x,y)}{\partial y^4}=\frac{w(y+2\Delta y)-4w(y+\Delta y)+6w(y)-4w(y-\Delta y)+w(y-2\Delta y)}{(\Delta y)^4}$$

$$2\frac{\partial^4 w(x,y)}{\partial x^2\partial y^2}=2\frac{w(x+\Delta x)-2w(x)+w(x-\Delta x)}{(\Delta x)^2}\frac{w(y+\Delta y)-2w(y)+w(y-\Delta y)}{(\Delta y)^2}$$

$$\frac{k}{D}w(x,y)=\frac{k}{D}w(x)=\frac{k}{D}w(y)$$

Adding the terms, the initial equation expressed by central differences have the form:

$$\begin{gathered}w_{i,j-2}\\+2w_{i-1,j-1}-8w_{i,j-1}+2w_{i+1,j-1}\\+w_{i-2,j}-8w_{i-1,j}+20w_{i,j}-8w_{i+1,j}+w_{i+2,j}\\+2w_{i-1,j+1}-8w_{i,j+1}+2w_{i+1,j+1}\\+w_{i,j+2}+h^4\frac{k}{D}w_{i,j}=h^4\frac{p_{i,j}}{D}\end{gathered}$$

This is the cell that will be run in the program in order to solve our problem. Geometrically, we can represent this cell according to the following figure:

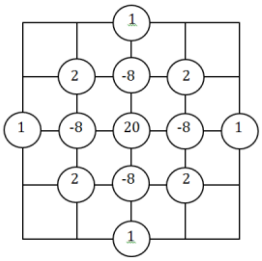

Fig.1. Schematic diagram of cell

A. Contour and Charging Conditions

The boundary conditions of our plate as well as the way the charge is distributed on it are expressed in the figure of the problem itself (figure 02). The lower edge AB and the upper edge DC are embedded, and the edges AD and BC are on simple support. The charge is uniformly distributed, but acts only on the lower half of the plate.

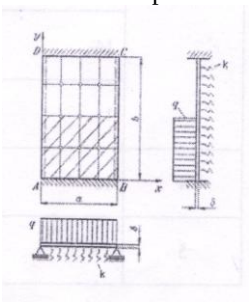

Fig. 2. Design of the proposed problem

It is important to note that the base cell arises from a fourth-order equation, and it is necessary for the calculation of the transverse displacement w two adjacent points in the i-direction and in the j-direction. Therefore, the artifice of "fictitious points" is necessary. The value of the dummy points are not necessarily zero. They follow the curvature of the surface and are therefore a function of the contour condition and the first edges analyzed. In simplified form, $w_{i-2,j}=w_{i,j}-w_{i-1,j}$ (support) and $w_{i-2,j}=w_{i,j}+w_{i-1,j}$ (clogged).

III. PROGRAM CONSTRUCTION

The program for cell implementation was developed in SciLab environment. Scilab is a high-level, interpretive programming language with numerous numeric tools, with many similarities to MatLab® [3]. The software is free, open source, does not require a license for use and installation, and can be distributed freely, so it is very much used in universities and other industrial applications. It was developed by the French consortium "Scilab", and it is constantly updated, especially with regard to new toolbox for various applications. The version used was 5.4.0, and can be downloaded at www.scilab.org/download.

To simplify, we can divide the program into four parts: (i) Acquisition of input values; (ii) construction of the Matrix of coefficients, called matrix A; (iii) construction of the matrix of independent terms, called matrix u; and (iv) calculations of the values of w. The results can be seen numerically, but it was opted to visualize by means of graphs able to translate in a more didactic way the behavior of the values reached.

Initially, it was not known the contour conditions, even the information being in the figure, ended up going unnoticed. So the program was created with the possibility of including the boundary conditions. It was preferred to maintain this functionality.

SciLab, like MatLab®, receives the input data by command lines. However, the x_mdialog function exists. With it, it is possible to open dialog boxes, requesting actions or adding values or strings, leaving the program more interactive. For example, in the program we have the lines, below, that generate the window of figure 03:

```
-->sig = x_mdialog('entre com os dados', txt, ['40';'40';'1']);
-->L = evstr(sig(1));
-->C = evstr(sig(2));
-->h  = evstr(sig(3));
```

The dialog box, which prompts you for board width, board length, and discretization value h. The values will be saved in the variables L, C and h, respectively. The same occurs with the boundary conditions, and the values of the elastic constant k, the load p, and the rigidity D.

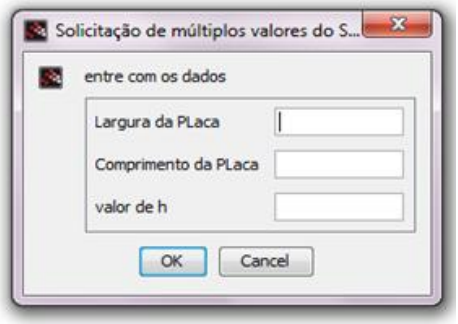

Fig. 3. Program Dialog Box

After the data entry procedures, the coefficient matrix A is constructed. This matrix has the coefficients of the cell, distributed in band along the matrix. For this, a routine was constructed capable of generating a line of coefficients, and it moves a column to each line. This is the largest array of the program. If the system is discretized in a grid of m columns, by n lines, we have a square matrix A of size mnXmn. The results will be detailed later with m = 40 and n = 40, so matrix A has dimensions of 1600x1600.

After the construction of the Matrix, we have in the next step the construction of the vector-column of independent values U. This construction takes into account the value of the boundary conditions, the data of the dummy points and the load q applied on the plate. Recalling that for values of i above half of the plate, the applied load is equal to zero. The base cell equation was applied to the development of these values. The initial construction took place in Matrix U1, then we aligned the terms of this matrix into a single column, called u. Finally, the most important point: the account itself, expressed by the line: **w = inv (A) * u**

At this point, a computational limit was realized to perform this operation, because matrices A slightly larger than 1600x1600 "burst" the variable stack memory, interrupting the inversion / multiplication operation, showing the error message expressed in figure 04. This 40x40 limit was found empirically.

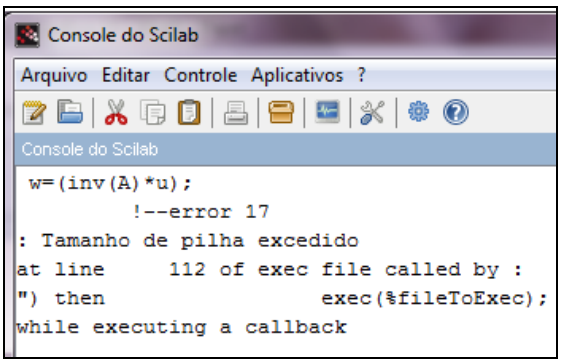

Fig. 04. Calculation memory exceeded, in discretizations greater than 40x40

With this, we already have the results of the values of w. These values could be expressed in a column-vector, or expressed in an array, positioned similarly to their location on the board. This matrix is the matrix V. The results were expressed in graphs, which have a better didactics of results analysis.

IV. RESULTS

The matrix V is created from the values of w. The results can be analyzed numerically, observing the matrix. However, an analysis in a relatively large array is not simple to view, and the smaller the discretization value h, the larger the array size, and the better the resolution and accuracy of the problem. So the option to graphically view the results.

For the purpose of examples, we will now adopt the following loading values (10^3), stiffness (10^6) and elastic constant (10^3), which are requested through the dialog box next to it, once the boundary conditions are entered.

A. Resolution x Discretization

The lower the value of h, the better the result, however, the greater computational power will be required to perform the calculations, because the higher the coefficient matrices, independent terms and also results. As already reported, a computational impediment lies in the memory capacity of the stack. The highest resolution was reached at 40x40. You can see below the improvement of the result with the discretization, in graph (figure 05):

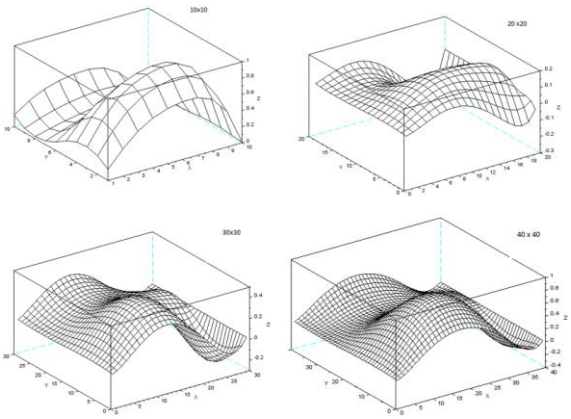

Fig. 5. Results of different discretizations

Thus, the best response occurs in the lower discretization. All the results of this point in day will be in grids of 40x40. The graphs of figure 5 were obtained with the mesh (V) command. In some cases, it is more convenient to express the results on contours, or equipotential surfaces. In Figure 06, the equivalent result is expressed in 2D graphs:

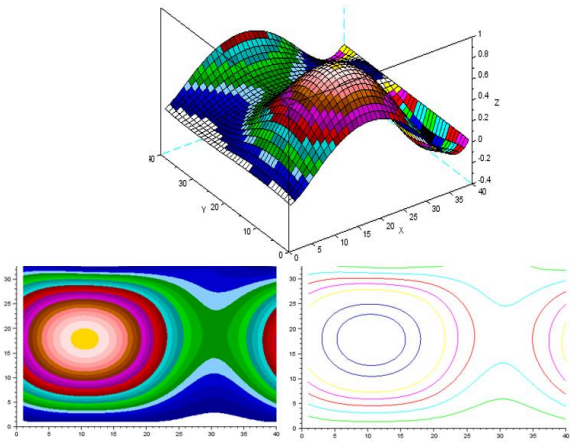

Fig. 06. Equivalent 2D chart and contour lines

B. Analysis of results

By analyzing the graphs, you can see the behavior of the board. Only half of the plate is loaded, the maximum deformation in the region close to the point (10,10). The region that is not subjected to the load also deforms due to the bimomento. By "climbing" on one side, we have the "get down" on the other side of the board.

We imagine some extreme situations to observe the behavior of the program and observe if the numerical / graph result matches the expected situation. For example: what is the behavior of the program if we enter p = 0 (no load). The result is expressed in figure 07:

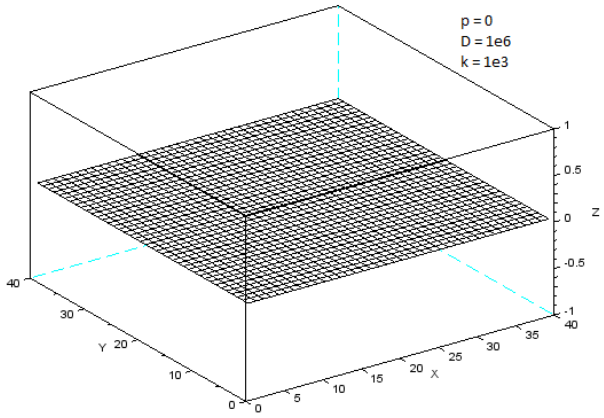

Fig. 7. Result for P = 0 (without load) - There is no transverse displacement

As expected, if there is no loading, there should be no deformation. The graphical response of the built program shows a simple flat plate with no deformation. Next, we insert in the variables, the value of the elastic constant as being zero. The result is shown in figure 8:

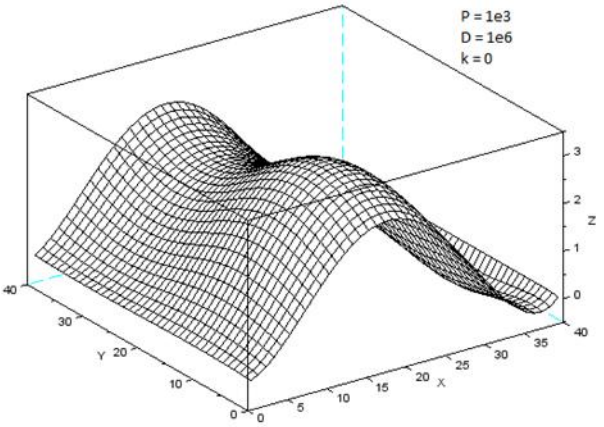

Fig. 8. Result for k = 0 (no load) - Increased strain amplitude

By making k = 0, we have the maximum deformation of the plate, under conditions p = 10^3 and D = 10^6. By increasing the value of k, we only reduce the amplitude of the deformation, keeping the deformation "standard" practically constant.

Now we can question what happens to the deformation by changing the rigidity of the plate, making it equal to zero. First of all, this is an IMPOSSIBLE situation, there is no way for a zero rigid board. In addition, in program calculations, the

value of D appears in the denominator, being mathematically impossible to divide by zero, generating an error in the program. But, just for comparative purposes, let's make a D small enough to realize the behavior of the graphical output of the program.

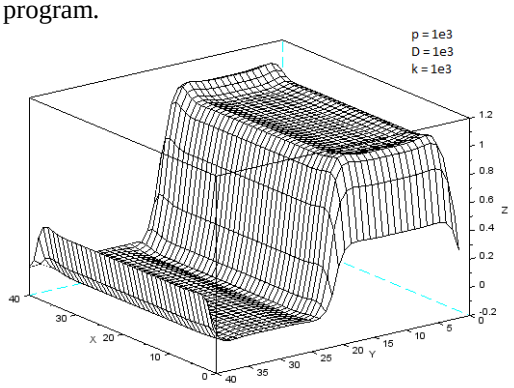

Fig. 9. Very small stiffness - high deformation, little interaction

It is easy to see that the structure has deformed following the load distribution along the half of the plate. Practically there was no interaction between an application point and its neighborhood.

With these punctual analyzes, it is possible to perceive that the result of the program corresponds to the corresponding physical situation. It is important to note that 40x40 square matrices were assembled. The program seamlessly runs non-square systems, for example a 20x60, or 10x70 card. The problem is that under these conditions, one dimension will have good resolution and the other will have low resolution. The best format found for viewing has been the one explored so far. In Figure 10, I highlight some results with non-symmetric plates (width ≠ length), for the same input data ($p = 10^3$, $D = 10^6$, $k = 10^3$).

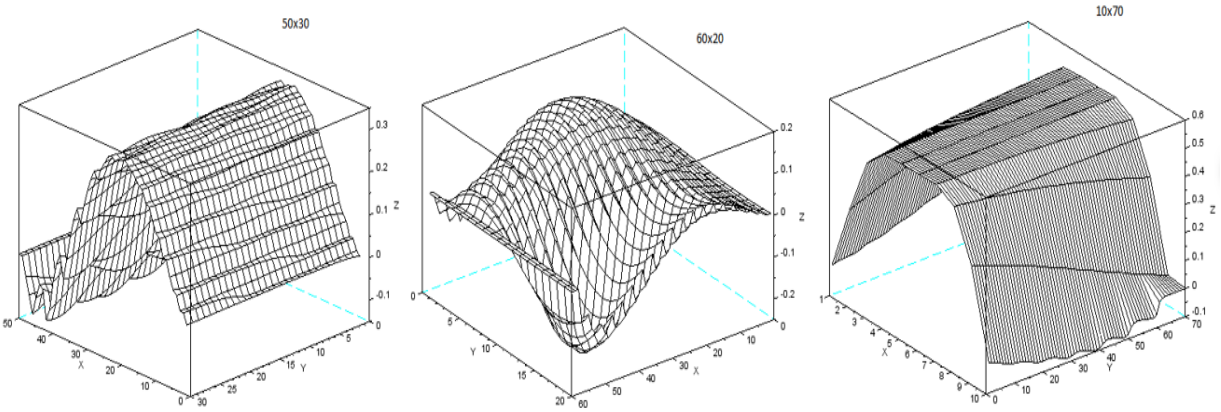

Fig. 10. Some results for non-symmetric plates (width ≠ length)

C. Convergence

A numerical method is said to converge when Δx and Δt tend to zero, and the numerical solution increasingly approaches the real solution [4]. The proposed problem has no time dependence, being a temporally static problem. Therefore, it automatically converges in time, since Δt = 0 in any situation. The Taylor series is convergent, since it is contained within the radius of convergence, which does not directly guarantee that the method converges. However, convergence was observed for all proposed situations.

V. CONCLUSION

The program developed fulfilled its objectives: to solve the problem proposed in the classroom and to provide a meaningful learning regarding the discipline of numerical methods, being the challenge to the level of the desired level of study in a graduation.

For the test situations, the program responded satisfactorily, achieved consistent results and proved versatile in its application. The numerical method is efficient in solving this type of problem. Only the software used did not immediately respond to more ostensive use, with even smaller discretizations.

REFERENCES

[1] KAMIŃSKI, Marcin. A generalized version of the perturbation-based stochastic finite difference method for elastic beams. Journal of Theoretical and Applied Mechanics, [S.l.], v. 47, n. 4, p. 957-975, Jan. 2009. ISSN 1429-2955.

[2] DOLICANIN C . B.,ET AL. Application of Finite Difference Method to Study of the Phenomenon in the Theory of Thin Plates. SER. A: APPL. MATH. INFORM. AND MECH. vol. 2,1 (2010), 29-43.2001. ISCAS 2001. The 2001 IEEE International

Symposium on, Sydney, NSW, 2001, pp. 511-514 vol. 5.

[3] Mayur Jain, Sheetal Bhande, Aditya Chhatre, Amit Naik, #Vishal Pande, *Prafulla Patil. CONTROL SYSTEM DESIGN USING OPEN SOURCE SOFTWARE (SCILAB). International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622. VNCET March 2012

[4] CHAPA, S. C.; CANALE, R. P. Métodos numéricos para engenharia. Tradução técnica: Helena Castro. São Paulo: McGraw-Hill, 2008.

Borges "Solving engineering problems using the finite difference method applied to Scilab." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 12, 2017, pp. 14-19.