A geometric model for the generation of models defined in Complex Systems

Y. Villacampa[1], F. J. Navarro-González[1] & J. Llorens[2]
[1]*Departamento de Matemática Aplicada, Universidad de Alicante, Spain*
[2]*Departamento de Sistemas Informático y Computación, Universidad Politécnica de Valencia, Spain*

Abstract

When studying and modelling Complex Systems where there can be a considerable number of variables and relations, it is very important to obtain methodologies that help us to model natural phenomena, these being phenomena present in the physical world and real life.

Hypothetically, if we consider a relation defined by a set of variables, $y = f(x_1, x_2, ..., x_n)$, symbolic and numerical methodologies can be obtained.

This article presents a model of n-dimensional finite elements that provides the basis for defining a numerical methodology for studying and modelling complex systems. The use of n-dimensional elements then allows us to represent the relation using the values of the same for a finite number of points, this being carried out by resolving an optimization problem.

To obtain the geometric model it was necessary to use the correct data structure design and programming to allow effective management of the acquisition and storage of the elements, the nodes considered for each, as well as the functions and procedures needed to approach the problem of optimisation.

Finally, we applied the methodology to determine the geometric model and the problem of optimization to study and model an environmental problem

Keywords: modelling, finite elements, Complex Systems.

1 Introduction

A complex system is considered to be any system where the number of variables and relationships that affect their evolution is high. In these systems, it is very important to acquire new methodologies that help us to model said relationships.

WIT Transactions on Ecology and the Environment, Vol 122, © 2009 WIT Press
www.witpress.com, ISSN 1743-3541 (on-line)
doi:10.2495/ECO090081

This allows us to model natural phenomena, these being phenomena present in the physical world and real life.

By considering a hypothetical relation defined by a set of variables, $y = f\left(x_1, x_2, ..., x_n\right)$, symbolic and numerical methodologies can be obtained.

In order to obtain models defined on the basis of the experimental data, there are several symbolic methodologies in scientific literature (Cortés et al. [1]; Splus [2]; Spss [3]; Villacampa et al. [4]; Verdú and Villacampa [5].) In said methodologies, the linear models are analysed in the same way, with the search for non-linear models being treated differently. With Splus and Spss, we can obtain a non-linear model every time programmes are run, as long as the modeller proposes a type of mathematical equation depending on certain parameters that are calculated on the basis of the experimental data. In the methodology developed in Cortés et al. [1]; Villacampa et al. [4] and implemented computationally, each time the programme is run families of models are generated, transforming them so that they can be treated in the same way as the linear models. In addition, Verdú and Villacampa [5], generates families of models in accordance with non-linear model types in the parameters. With regard to numerical methodologies, the article Pérez-Carrió et al. [6] develops a numerical methodology to be applied in two dimensions, using finite elements.

To study complex systems such as natural systems that include the material and physical world, such as an ecosystem, we need to develop methodologies that allow us to introduce greater complexity and which can then be used in hybrid fashion with the above-mentioned methodologies.

Acquisition of a methodology for $n \geq 2$ has to be computationally valid for large values of *n*. This means that we have to treat the problem differently to the method developed for *2-dimensional* finite elements (Pérez-Carrió et al. [6]) both as regards the methodology and the algorithm. This is why the writers start by defining a geometric model of *n-dimensional* finite elements in such a way that it allows enhanced management of use and storage when implemented computationally.

2 Geometric model of *n-dimensional f*inite elements

For any natural number, we define a domain contained in $\Re^n$, in which we define a family of finite elements recurrently.

For example, with a natural number $d \in N$, the domain is defined as the subset $D \subset \Re^d$ defined by the hypercube $D = [0,1]^d$.

2.1 Discretization

For each edge we are going to consider discretizations with the same number of elements, as well as the same number of nodes. If the number of elements for each edge is c, the number of nodes defined is $c+1$. Therefore, the model

defined will consist of a total of c^d elements and $(c+1)^d$ nodes. From now on, we shall denominate the value of c as the complexity, this being the number considered initially to generate discretization in finite elements.

For a generic element, we shall define a set of 2^d nodes per element, so that 2^d elements coincide in an internal node.

The need to consider relations, $y = f(x_1, x_2, ..., x_n)$ of a certain complexity leads us directly to consider a geometric model that is complex in the sense of the number of nodes and elements, first having to modify the methodology normally used in $\Re^2$ [3] to enhance the computational implementation of the model.

2.2 Elements and nodes in the geometric model

For the domain $D = [0,1]^d = [0,1]x[0,1]x... \xrightarrow{d} ...x[0,1]$, we decided to deal with the numbering of both elements and nodes by starting the variation more quickly in the first interval, then the second and so on.

2.2.1 Global representation of elements and nodes

We now define two equivalent representations that offer better management of the elements and nodes for which we consider global numbering starting at "0".

Definition

For an element whose global numbering is defined by M, with c being the complexity defined, its representation is defined by a multi-index that gives us its position and is calculated by expressing M in c basis, but ordering the coefficients in the opposite direction to normal.

For example, with $D = [0,1]^3$, if we consider $c = 4$ and the element with global numbering of $M = 6$, then $M = (2,1,0)$.

Definition

For a node whose global numbering is defined by M, with c being the complexity defined, its representation is defined by a multi-index that gives us its position and is calculated by expressing M on the basis of $c+1$, but ordering the coefficients in the opposite direction to normal l.

Definition Directional Multi-Index (DMI)

A DMI is any multi-index that can be written as:

$$MID = \left\{ \left(j_1, j_2, j_3, ..., j_d \right) / \, j_i = 0,1 \right\}_{j=1}^{2^d}$$

Using the representation of each multi-index in 2 basis, we can also obtain global numbering of the directional multi-indexes $\{m_i\}_{i=1}^{2^d}$.

2.2.2 Determination of the nodes associated with an element

In order to determine which nodes are associated with an element we simply need to start from the node closest to the origin or node "0" and add the multi-indexes denominated as directional multi-indexes. In addition, the methodology defined above directly defines the node closest to origin, as it corresponds to the multi-index associated with the element expressed in $c+1$ basis.

For a generic element, we define a family of interpolation functions or form functions equal to the number of nodes in the element.

$$\{N_i\}_{i=1}^{2^d} = \left\{ N_{m_i}(s_1, s_2, ..., s_d) \right\}_{m_i \in MID}$$

2.2.3 Coordinates

For an element, we consider the global coordinates of the nodes associated with said element, $\{D_{m_i}\}_{m_i \in MID}$. With $(e_1, e_2, ..., e_n)$ also being the coordinates of the node closest to origin, then its coordinates are $D_{m_i} = (e_1, e_2, ..., e_n) + h\, m_i$, $\left(D_{m_i} = \left(D_{m_i 1}, D_{m_i 2}, ..., D_{m_i k}, ...D_{m_i n}\right)\right)$, where h represents the size or length of the elements in each sub-interval.

Throughout this article h is called the *"discretization size"* or "*mesh refinement"*.

Definition

For a point of global coordinates $(x_1, x_2, ..., x_n)$ found in an element whose closest node to origin is $(e_1, e_2, ..., e_n)$, the relative coordinates are defined as, $(x_{R1}, x_{R2}, ..., x_{Rn}) = (x_1 - e_1, x_2 - e_2, ..., x_n - e_n)$, with $x_{Rj} \in [0, h]$.

Definition

The system of local coordinates, $(s_1, s_2, ..., s_n)$, with $s_i \in [-1, 1]$, is defined by,

$$s_k = \frac{2}{h} x_{Rk} - 1$$

Therefore, it can be expressed:

$$x_k = e_k + \frac{h}{2}(s_k + 1)$$

$$x_k = \sum_{m_i \in MID} N_{m_i}(s_1, s_2, ..., s_d)\, D_{m_i k}$$

3 Computational implementation of the geometric model

The first step towards obtaining the new complex methodology is the generation of a geometric model of n-dimensional finite elements. Using n-dimensional finite elements allows us to represent the relation on the basis of the values of the same at a finite number of points, through resolution of an optimization problem.

To obtain the geometric model, it was necessary to use data structure design and programming that allowed us the effectively manage the acquisition and storage of the elements, the nodes considered for each and the functions and procedures used to approach the optimization problem.

3.1 Numerical methodology

The geometric model defined above is applied to generate a new modelling methodology. We start with an a priori relation between a set of variables defined by $y = f\left(x_1, x_2, ..., x_d\right)$ and of which we know a set of experimental data.

$$\left\{y_j\right\}_{j=1}^{n} \quad \left\{P_j\right\}_{j=1}^{n} / P_j = \left(a_{1j}, a_{2j}, ..., a_{jd}\right)$$

We can define a model function generated from its values at the nodes defined in the geometric model, $\vec{u} = \left(u_o, u_1, ..., u_{(c+1)^d}\right)$ which we represent at a point P by $FEM\left(P\right) = \overrightarrow{N\left(P\right)} \bullet \vec{u}$.

The vector $\vec{u} = \left(u_o, u_1, ..., u_{(c+1)^d}\right)$ is obtained from the experimental data and with the condition that the following error function is minimized.

$$E = \sum_{j=1}^{NP} \left(FEM\left(P\right) \bullet \vec{u} - z_j\right)^2 \quad (*)$$

The authors called this problem (*) the optimization problem.

The dependence of the model on the experimental data leads to an optimization problem that causes problems when the data is insufficient and the equation system resulting from seeking the minimum is also a system with more than one solution. To these difficulties we have to add that arising from its very complexity.

3.2 Computational algorithm to generate the geometric model

When studying the computational implementation of the geometric model underpinning the new methodology, we opted to use an abstract global approach within the framework of object-oriented programming (OOP).

Said approach has the advantage of a broad focus that allows us to deal with problems other than the original and take advantage of all the code implemented.

The analysis was carried out using two large blocks of code:

- Block I: File read and input formats interface.
- Block II: Generator of the geometric model and the equation system.

For Block I, corresponding to the file read and input formats interface, we programmed a single library. This includes the functions and procedures required to read the data in different formats (plain text file, Excel file and XML file), as well as transforming the data to the hypercube $D = [0,1]^d$. It also includes the generation of "*.log"* reports used to execute the application.

Block II corresponds to the generator of the geometric model presented in this article and the equation system generator that will solve the optimization problem and allow us to obtain a new modelling methodology. Two libraries were programmed for this block: *plantilla_problema* and *lib_fem.*

The *plantilla_problema* library defines the following objects:

- *TProblema* – Generates a dynamic data matrix in which the experimental data is stored. It also stores all information associated with the data file routes. One of the properties of this object references the structures needed to store the discretization of the domain $D = [0,1]^d$ in a finite family of finite elements.

The *lib_fem library* defines the following objects:

- *TElemento* – For a generic n-dimensional element, in this class we store the information corresponding to an element and its nodes. We define the descendant class, *TElemento_NDCuadrado* for using an n-dimensional hypercubic *TElemento*.
  It contains the functions required to carry out transformations between the representation in directional multi-indexes DMI and the global index.

- *TMesh* – A generic mesh containing the set of elements and nodes ordered by their global indexes.

- *TDiscretización* – Represents the finite element model defined for the whole domain. This means that this object stores the mesh information obtained from discretization of the domain. As descendant of this object, we implement the *TDiscretización_NDCuadrado* class, this being the class associated with discretization in hypercubes. It is a representation of the generic element $[0,1]^d$, which includes methods needed to calculate the form functions of a point in local coordinates, as well as the functions for the change between the local, relative and global coordinates of a point.

The geometric model: algorithm

The algorithm for constructing the geometric model can be broken down into three main operations. These are:

1. File reading
2. Construction of the geometric model

The more detailed analysis of each of these operations is as follows:

Define the library *lib_fem*

Define the object

File reading

open input file

read number of rows, n_f

read number of columns, n_c

Define independent variable data matrix
Define dependent variable data matrix
close input file

If the variables are not normalised, normalise the values to $D = [0,1]^d$

Generation of the geometric model

Define refinement $h = 1/c$

Creation of Mesh object
For i=1 to number of nodes do
Generation of node i
Acquisition of the multi-index of the node
Calculation of the global coordinates of the node
Store the node in the Mesh object: TMesh
End loop
For i=1 to number of nodes do
Generation of element i
Acquisition of the multi-index of the element
Calculation of the nodes associated with an element
Store each node in TElement
Store the element in the Mesh object
End loop

3.3 Methodology of the modelling

With the previously generated geometric model, we generate the algorithm that allows us to resolve the numerical methodology defined in Section 3.1 The first step in said resolution is the generation of an equation system that is implemented computationally as follows.

Define system matrix
For i= 1 to number of nodes do
Calculation of local coordinates
Determination of the current element
Calculation of the form functions at the point
Modification of the system matrix
End loop.

4 Applications

Finally, the previously developed methodology was applied by determining the geometric model for the data relating to soya bean cultivation in Azul, Buenos Aires Argentina and a study was carried out of the phenological stage from sowing to flowering. In this case the authors have resolved the optimization problem comparing the results with the model obtained symbolically. In addition, the geometric model was applied to a set of data allowing *3D* visualisation.

4.1 A phenological model from the soybean

It has been considered data from soybean cultivation in Asgrow 4656, to quantify the effect of the sum of temperatures and the photoperiod on the duration of the phonological period from sowing to flowering. The first data come from years 1997-1998 and the results are compared with the models obtained with the symbolic methodology developed in Verdú and Villacampa [5]. The second set of data are compared using the lineal model published in Confalone et al. [7] for the years 1997–1999, where the average sum of temperatures and the photoperiod are considered.

4.1.1 Geometric model

In both cases the geometric model would be:
The first step it is the selection of the complexity.
With this complexity, the program generates the mesh, obtaining the set of nodes, elements and the relations between them.

4.1.2 Cultivar Asgrow 4656, 1997-98

In this case it has been possible to solve the problem of optimization and to get the finite element model, to compare the methodology with the results obtained by the methodology developed in Verdú and Villacampa [5] that gives the next equations:

$$DDS = 0.098\ ST + 7.954\ F + 117.75 \qquad R^2 = 0.99$$

$$DDS = \left(0.00125\ ST - 5.501\right)^3 + \left(0.1593\ F + 10.5434\right)^2 \qquad R^2 = 0.99$$

Comparing the errors between the estimated values on the points, obtained by these equations and the finite element method, and the experimental ones, it is clear that the results are similar or even better in many points.

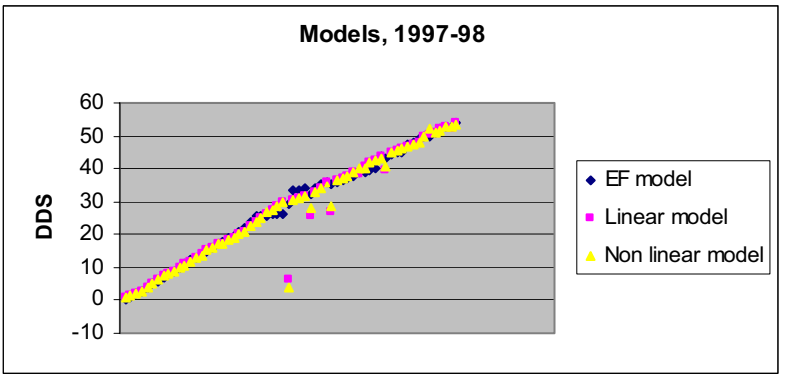

Figure 1: EF model, linear model and non-linear model 1997–1998.

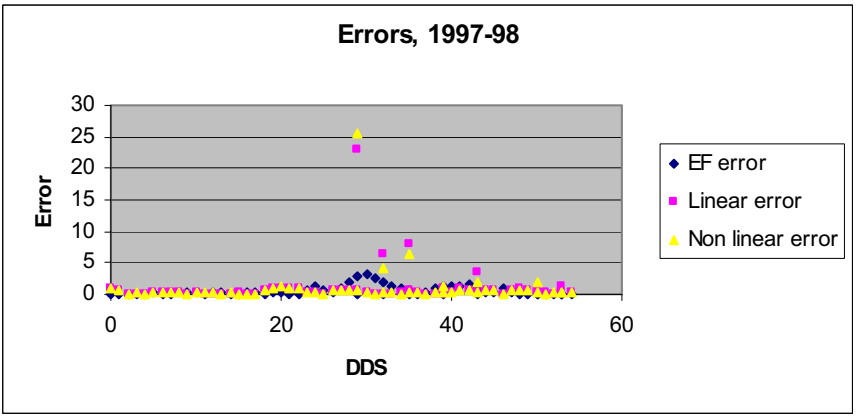

Figure 2: Errors, 1997–1998.

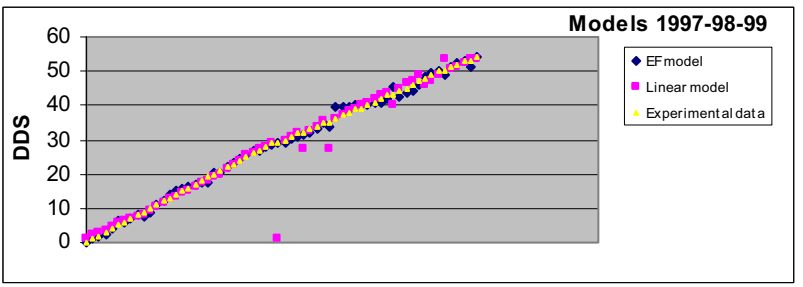

Figure 3: EF model and linear model, 1997–1999.

4.1.3 Cultivar Asgrow 4656, 1997–1999

In this case it has been possible to solve the problem of optimization and to get the finite element model, to compare the methodology with the linear model (Confalone et al. [7]):

$$DDS = 0.106\ ST - 0.072\ F + 1.7006 \quad R^2 = 0.99$$

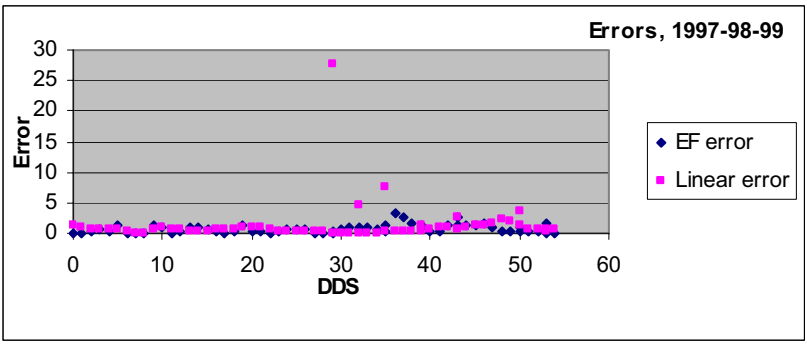

Figure 4: Errors, 1997–1999.

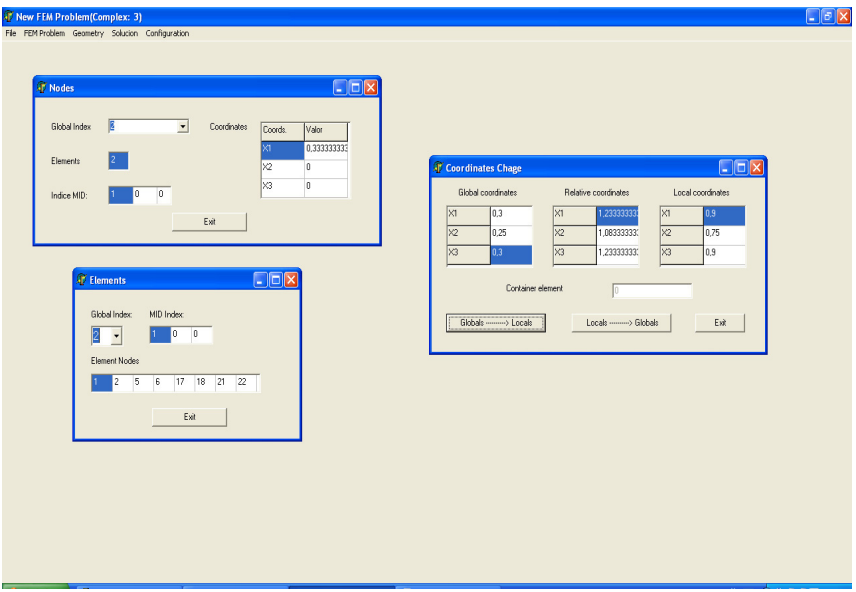

Figure 5: Nodes, elements and coordinates change.

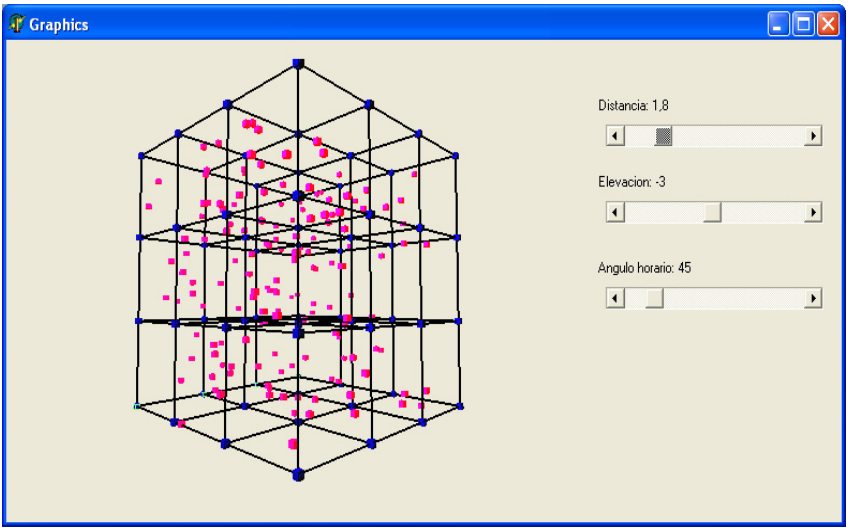

Figure 6: Geometric model.

4.2 Geometric model in *3D*

The data of this model are obtained from a randomly generated set of points in the domain $[0,1]^3$ using a function, $f(x,y,z) = x + 2y + z^2$, to get the values and distorting them with a normal error.

The 3-dimensional model is presented in the next graphics where the program main window and the data and discretization are shown.

5 Conclusions and future research

The first step in obtaining the new methodology for studying and modelling complex systems involves generating a geometric model of n-dimensional finite elements. The use of n-dimensional elements then allows us to represent the relation, $y = f(x_1, x_2, ..., x_n)$, using the values of the same at a finite number of points, and from the resolution of an optimization problem (*).

When obtaining the geometric model, it was necessary to design and programme the data structures correctly, so as to allow effective management of the acquisition and storage of the elements, the nodes considered for each and the functions and procedures used to deal with the optimization problem (*).

Future research should focus on obtaining a method that effectively resolves the optimization problem (*) defined by an equation system for the case of $n \geq 2$. This is due to the dependence of the experimental data function and the complexity defined in the geometric model, which leads to the resolution of the optimization problem (*) and that of the equation system presenting problems due to inadequate data and the multiplicity of the solution.

Acknowledgement

The authors would like to express their thanks to Dra. A. Confalone, who allowed access to the experimental data used in [7].

References

[1] Cortés, M., Villacampa, Y., Mateu, J., Usó, J.L. (2000). *A new methodology for modelling highly structured systems.* Environmental Modelling & Software (1364-8152). V.**15**, nº 2, pp. 461- 470.
[2] Splus. (1997). MathSoft,Inc. Seattle, Washington. USA.
[3] Spss. (1999). Inc. Software. Chicago. USA.
[4] Villacampa, Y., Cortés, M.., Vives, F., Usó, J.L., Castro, M. A. (1999). *A new computational algorithm to construct mathematical models.* Advances in Ecological Sciences. Ecosystems & Sustainable Development. Serie II. 185312 687X – pp. 323- 332.
[5] Verdú,F., Villacampa,Y. (2008). A Computational algorithm for the multiple generation of nonlineal mathematical models and stability study. Advances in Engineering Software. V**39**, Issue 5, 430-437.
[6] Pérez-Carrió, A.,Villacampa,Y., Llorens, J. (2009). A computational algorithm for system modelling based on bi-dimensional finite element techniques. Advances in Engineering Software. V **40,** nº 1, pp. 30-40.
[7] Confalone, A., Villacampa, Y., Reyes, J.A., García-Alonso, F., Verdú, F. (2007). A phonological model for soybean. Ecosystems and Sustainable Development VI. 978-1-84564-088-0, pp: 81-88.