

Teaching and Learning Iterative Methods For Solving Linear Systems Using Symbolic and Numeric Software

E. A. CARIAGA,¹ M. C. NUALART²

¹*Departamento de Ciencias Matemáticas y Físicas, Universidad Católica de Temuco, Casilla 15-D, Temuco, Chile*

²*Escuela de Informática, Universidad Católica de Temuco, Casilla 15-D, Temuco, Chile*

Received 23 February 2001; accepted 1 May 2002

ABSTRACT: An integral approach is presented to strengthen the teaching and learning processes in the environment of the undergraduate course Numerical Analysis (NA) for engineering, examining the advantages of combining the symbolic and numeric paradigms. In particular, the methodology is illustrated with the iterative methods: Gauss–Seidel (GS) and Conjugated Gradient (CG), for the numeric solution of Linear Systems (LS). The computer tools MATLAB and MAPLE are used in a pedagogic model that requires the explicit definition of Prospective Learnings and Activities of Learning. © 2002 Wiley Periodicals, Inc. *Comput Appl Eng Educ* 10: 51–58, 2002; Published online in Wiley InterScience (www.interscience.wiley.com.); DOI 10.1002/cae.10008

Keywords: linear systems; conjugated gradients methods; symbolic and numeric computation tools; MAPLE; MATLAB

INTRODUCTION

The undergraduate curriculum in engineering has traditionally included a course of Numerical Analysis

(NA), which is often located at the end of the mathematical line and at the beginning of courses in sciences of the engineering, constituting in a bridge between the basic sciences and the sciences of the engineering.

The NA is part of the applied mathematics, and it is intimately related with scientific computing. For this reason its teaching has incorporated for several decades the use of a programming language, such as, Fortran IV, Fortran 77, Fortran 90, C, Pascal, etc . . . , for the computational implementation of the numeric

Correspondence to: E. A. Cariaga (ecariaga@uct.cl).

Contract grant sponsor: Direction of Investigation (Catholic University of Temuco; through the project DIUCT); contract grant number: DIUCT 2000-4-02.

© 2002 Wiley Periodicals, Inc.

algorithms. However, the generation of such computational tools as MAPLE, DERIVE, MATHEMATICA, MATLAB, MATHCAD, to mention some, has motivated a deep revision of the traditional teaching ways in engineering, allowing the students to concentrate more on the design, and less on routine calculation procedures.

In this article, an integral approach is presented to strengthen the teaching and learning processes in the environment of the undergraduate course on NA for engineering, examining the advantages of combining the symbolic and numeric paradigms [1]. In the first section, we review the importance of the NA in the making of an engineer. In the second section, we present the iterative methods: Gauss–Seidel (GS) and Conjugated Gradient (CG), for the numeric solution of Linear Systems (LS). Methods of the type CG, are not usually incorporated in undergraduate courses, however, we show that the merged use of MAPLE and MATLAB allow the intuitive understanding of their theoretical foundations and their practical use in the solution of LS [2]. In the third section, we presented a teaching model and learning that it incorporates varied learning experiences for the students based on the use of MAPLE and MATLAB. The excellent thing is that the use of such tools as MAPLE and MATLAB turn out not to be useful alone just to strengthen the teaching of the discipline but also as learning instruments.

NUMERICAL ANALYSIS AND ENGINEERING

Accreditation Board for Engineering and Technology, U.S. (ABET) developed Engineering Criteria 2000 [3]. The criteria state that engineering programs must demonstrate that their graduates have, among others, an ability to apply knowledge of mathematics, science and engineering, an ability to design and conduct experiments as well as to analyze and interpret data, and an ability to design a system, component, or process. The importance of an appropriate mathematical formation is clearly established [4].

The Siam Report [5] shows that in non-academic organizations, the mathematics is present in such activities as: research and development of mathematical tools and algorithms, creation and support of mathematical and computational techniques associated with a specific product or service, and consulting or modeling for internal or external customers. The mathematical functions of greatest value were characterized by managers as modeling and simulation, mathematical formulation of problems, algo-

rithm and software development, problem-solving, statistical analysis, verifying correctness, and analysis of accuracy and reliability. We also highlight in Table 4 of Ref. [5] (indicates selected associations between areas of mathematics and applications encountered in the site visits) the Numerical Analysis turned out to be essential in all the applications.

It clearly establishes the relevance of evaluating the use of computational resources in the environment of the teaching and of the learning of the NA, specifically, the iterative solution of LS.

ITERATIVE METHODS FOR SOLVING LINEAR SYSTEMS

In recent years, much research have focused on the efficient solution of large sparse or structures linear systems using iterative methods [6–8].

Small matrices, say with dimension 3 or 30, may arise directly with more or less arbitrary entries in scientific problems as representations of the relations between three forces in a structure, perhaps, or between 30 species in a chemical reaction. Large matrices, by contrast, usually arise indirectly in the discretization of differential or integral equations. The most obvious structure of a large matrix is sparsity, i.e., preponderance of zero entries. For example, a finite difference discretization of a partial differential equation may lead to a matrix of dimension $n = 100,000$ with only 10 non-zero entries per row. This kind of structure is readily exploited by the iterative methods. The methods described are most often useful for very large sparse or structured matrices, for which direct methods are too costly in terms of computer time and/or storage.

The relative chapter to the iteratives methods to solve LS in a standard course of NA considers the methods of Jacobi, Gauss–Seidel, SOR, and possibly SSOR [9,10]. On the other side, the numeric solution of partial differential equations requires the use of modern iterative methods, such as, Preconditioned Conjugate Gradients Method (PCG), BiConjugate Gradients Method (BICG), BiConjugate Gradients Stabilized Method (BICGSTAB), Conjugate Gradients Squared Method (CGS), Generalized Minimum Residual Method (GMRES), Minimum Residual Method (MINRES), Quasi-Minimal Residual Method (QMR), and Symmetric LQ Method (SYMMQL), (see directory *sparfun* in MATLAB), that are not treated in the undergraduate course of NA. We postulate the incorporation of the method CG, to the standard course of NA, like an entrance door to the modern iterative methods.

Iterative Method of Gauss–Seidel

The problem is to solve the LS: $\mathbf{Ax} = \underline{b}$, where \mathbf{A} is an $n \times n$ non-singular matrix and \underline{b} is a given n -vector. If $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$, where, \mathbf{L} is the strict inferior triangular matrix of \mathbf{A} , \mathbf{D} is the diagonal matrix of \mathbf{A} and \mathbf{U} is the strict upper triangular matrix of \mathbf{A} , the LS: $\mathbf{Ax} = \underline{b}$ is equivalent to $(\mathbf{L} + \mathbf{D} + \mathbf{U})\underline{x} = \underline{b}$, i.e., $(\mathbf{L} + \mathbf{D})\underline{b} = -\mathbf{U}\underline{x} + \underline{b}$, i.e., $\underline{x} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}\underline{x} + (\mathbf{L} + \mathbf{D})^{-1}\underline{b}$. Starting from this last equality is defined the iterative succession of GS:

$$\begin{aligned}\underline{x}^{(l+k)} &= -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}\underline{x}^{(k)} + (\mathbf{L} + \mathbf{D})^{-1}\underline{b}, k \\ &= 0, 1, 2, \dots\end{aligned}$$

which is convergent if \mathbf{A} is symmetric and definite positive.

Program 1, which is a *m-file* of MATLAB, solve the LS: $\mathbf{Ax} = \underline{b}$, by mean of Gauss–Seidel method.

Program 1

```
function x = gs(A, b)
D = diag(diag(A))
U = triu(A) - D
L = tril(A) - D
x0 = zeros(size(b));
x = x0
xb = x - 999
n = 0
while norm(x - xb, inf) > 1e - 30
xb = x
x = -inv(L + D)*U*x + inv(L + D)*b
n = n + 1
if n > 300 break end
end
end
```

Conjugate Gradient Method

The content of this section is based on Ref. [10]. We consider the problem of find \underline{x} , such that, $\mathbf{Ax} = \underline{b}$, if $\mathbf{A}^T = \mathbf{A}$ and $\underline{x}^T \mathbf{Ax} > 0, \underline{x} \neq 0$. The Fundamental Lemma enunciates the fundamental property that allows to build the CG method and related.

Fundamental Lemma. *If \mathbf{A} is symmetric and positive definite, the problem to solve $\mathbf{Ax} = \underline{b}$ is equivalent to the problem of minimizing the quadratic form $q(\underline{x}) = \langle \underline{x}, \mathbf{Ax} \rangle - 2\langle \underline{x}, \underline{b} \rangle$, where $\langle \cdot, \cdot \rangle$ is a scalar product defined by $\langle \underline{x}, \underline{y} \rangle = \underline{x}^T \underline{y}$.*

Proof. The behavior of the function $q(\underline{x})$ is examined along a ray unidimensional. Consider $\underline{x} + t\underline{v}$, with $\underline{x}, \underline{y}, \underline{v}$ vectors and t a scalar. A direct com-

putation shows that for all scalar t : $q(\underline{x} + t\underline{v}) = q(\underline{x}) + 2t\langle \underline{v}, \mathbf{Ax} - \underline{b} \rangle + t^2\langle \underline{v}, \mathbf{Av} \rangle$, because \mathbf{A} is symmetrical. On the other hand, the t value that gives us the minimum point is; evaluating $q(\underline{x} + \hat{t}\underline{v})$ in \hat{t} , $q(\underline{x} + \hat{t}\underline{v}) = q(\underline{x}) - \langle \underline{v}, \underline{b} - \mathbf{Ax} \rangle^2 / \langle \underline{v}, \mathbf{Av} \rangle$. The calculation previous sample that when happening of \underline{x} to $\underline{x} + \hat{t}\underline{v}$ there is always a reduction in the value of $q(\underline{x})$, unless \underline{v} be orthogonal to the residual, i.e., $\langle \underline{v}, \underline{b} - \mathbf{Ax} \rangle = 0$, starting from that which the equivalence presented in the Fundamental Lemma is deduced easily.

The previous proof suggests the following iterative method to solve $\mathbf{Ax} = \underline{b}$: $\underline{x}^{(l+k)} = \underline{x}^{(k)} + t_k \underline{v}^{(k)}$, where, $t_k = \langle \underline{v}^{(k)}, \underline{b} - \mathbf{Ax}^{(k)} \rangle / \langle \underline{v}^{(k)}, \mathbf{Av}^{(k)} \rangle$.

A natural election of search direction $\underline{v}^{(k)}$ is minus gradient of $q(\underline{x})$ evaluate in $\underline{x}^{(k)}$, which points in the direction of the residual $\underline{r}^{(k)} = \underline{b} - \mathbf{Ax}^{(k)}$. Therefore, the residuals generated in each iteration could be chosen as search addresses. This first possibility gives this way origin to those called Methods of Steepest Descent. However, this type of methods are rarely used in this class of problem due to their slowness. The conjugated gradient method belongs to a family whose members share the strategy of minimizing $q(\underline{x})$ along a succession of rays. The search direction that characterize the CG method are specified in the Fundamental Definition.

Fundamental Definition. (a) *If \mathbf{A} is a matrix $n \times n$ definite positive, a set of vectors $\{\underline{u}^{(1)}, \underline{u}^{(2)}, \dots, \underline{u}^{(n)}\}$ it is said \mathbf{A} -orthonormal if and only if $\langle \underline{u}^{(i)}, \mathbf{A}\underline{u}^{(j)} \rangle = \delta_{ij}, 1 \leq i, j \leq n$.*

(b) *If \mathbf{A} is a matrix $n \times n$ definite positive, a set of vectors $\{\underline{v}^{(1)}, \underline{v}^{(2)}, \dots\}$ it is said \mathbf{A} -orthogonal if and only if $\langle \underline{v}^{(i)}, \mathbf{A}\underline{v}^{(j)} \rangle = 0, i \neq j$.*

Notice that of a \mathbf{A} -orthogonal system, it is possible to obtain a \mathbf{A} -orthonormal system by way of a normalization process given by $\underline{u}^{(i)} = \underline{v}^{(i)} / \|\underline{v}^{(i)}\|_A$, where $\|\underline{x}\|_A^2 = \langle \underline{x}, \underline{x} \rangle_A = \langle \underline{x}, \mathbf{Ax} \rangle = \underline{x}^T \mathbf{Ax}$.

The CG method is preferable to the gaussian elimination when \mathbf{A} it is large and sparse. Theoretically the algorithm CG will give us the solution of the system $\mathbf{Ax} = \underline{b}$ in n steps at the most. We should not expect from a iterative method to obtain the solution with absolute precision after n steps. In fact, it is expected to obtain a satisfactory answer in less than n steps for extremely big systems. In well conditioned problems, the number of necessary iterations, for that the CG method converges satisfactorily, can be much smaller than the order of the system. These considerations are based in the following Fundamental Theorem.

Fundamental Theorem. *Suppose \mathbf{A} is an $n \times n$ positive definite matrix.*

- a) Let be $\{\underline{u}^{(1)}, \underline{u}^{(2)}, \dots, \underline{u}^{(n)}\}$ an **A**-orthonormal system.
 Define $\underline{x}^{(i)} = \underline{x}^{(i-1)} + \langle \underline{b} - \underline{Ax}^{(i-1)}, \underline{u}^{(i)} \rangle \underline{u}^{(i)}$, $1 \leq i \leq n$, where, $\underline{x}^{(0)}$ is an arbitrary point of \mathbb{R}^n .
 Then $\underline{Ax}^{(n)} = \underline{b}$.
- b) Let be $\{\underline{v}^{(1)}, \underline{v}^{(2)}, \dots, \underline{v}^{(n)}\}$ an **A**-orthogonal set of vectors not null.
 Define $\underline{x}^{(i)} = \underline{x}^{(i-1)} + \frac{\langle \underline{b} - \underline{Ax}^{(i-1)}, \underline{v}^{(i)} \rangle}{\langle \underline{v}^{(i)}, \underline{Av}^{(i)} \rangle} \underline{v}^{(i)}$, $1 \leq i \leq n$, where, $\underline{x}^{(0)}$ is an arbitrary point of \mathbb{R}^n . Then $\underline{Ax}^{(n)} = \underline{b}$.

Another distinctive property of the CG method is that the residuals form a system orthogonal in the ordinary sense, that is to say, $\langle \underline{r}^{(i)}, \underline{r}^{(j)} \rangle = 0$, $i \neq j$.

In the Table 1, Algorithm 1, [10], is a formal version of the CG method that incorporates the previously established characteristics. On the other hand, the Algorithm 2, [10], is a modification of the Algorithm 1 that allows an implementation more efficient computational, while the third column, titled Program 2, [2], is a *m-file* of MATLAB starting from the Algorithm 2.

MODEL OF TEACHING AND LEARNING

The Model of Teaching and Learning consists of the definition (to-priori) of Prospective Learnings and the design of Activities of Learning. The Prospective Learning express the capabilities and competitions that is desired that the students achieve. They guide the pedagogic process and they give a direction to

the learning process. They are determinant to define the evaluation approaches. The Learning Activities are actions and processes that the students have to live to achievement of the Prospective Learnings. In this sense such tools as MAPLE and MATLAB are powerful resources for the design and implementation of Activities of Learning.

Contents

The considered topics are the GS and CG methods in the numeric solution of system of lineal equation context. Previously, the students have to know Linear Algebra.

Prospective Learnings

We propose the following prospective learnings:

- 1) They distinguish conceptually between direct and iterative methods to solve LS.
- 2) They know the definition and properties of the following iterative methods: Jacobi, Gauss–Seidel, SOR, SSOR and CG. However, in this article we will only make mention to GS and CG methods.
- 3) They understand the conceptual foundations of the family of methods of the type Conjugated Gradient.
- 4) They outline problems that involve the iterative solution of LS.
- 5) They apply iterative procedures to solve LS.

Table 1 Conjugate Gradient Method

Algorithm 1	Algorithm 2	Program 2
input $\underline{x}^{(0)}, M, a, \epsilon, eps$ $\underline{r}^{(0)} = \underline{b} - \underline{Ax}^{(0)}$ $\underline{v}^{(0)} = \underline{r}^{(0)}$ output $0, \underline{x}^{(0)}, \underline{r}^{(0)}$ for $k = 0, 1, \dots, M-1$ do if $\underline{v}^{(0)} = 0$ then stop $t_k = \langle \underline{r}^{(k)}, \underline{r}^{(k)} \rangle / \langle \underline{v}^{(k)}, \underline{Av}^{(k)} \rangle$ $\underline{x}^{(1+k)} = \underline{x}^{(k)} + t_k \underline{v}^{(k)}$ $\underline{r}^{(1+k)} = \underline{r}^{(k)} - t_k \underline{Av}^{(k)}$ if $\ \underline{r}^{(1+k)} \ ^2 < eps$ then stop $S_k = \langle \underline{r}^{(1+k)}, \underline{r}^{(1+k)} \rangle / \langle \underline{r}^{(k)}, \underline{r}^{(k)} \rangle$ $\underline{v}^{(1+k)} = \underline{r}^{(1+k)} + S_k \underline{v}^{(k)}$ output $1+k, \underline{x}^{(1+k)}, \underline{r}^{(1+k)}$ end	input $x, M, A, b, \epsilon, eps, del$ $r = b - Ax$ $v = r$ $c = \langle r, r \rangle$ for $k = 1, \dots, M$ do if $\langle v, v \rangle^{1/2} < del$ then stop $z = Av$ $t = c / \langle v, z \rangle$ $x = x + tv$ $r = r - tz$ $d = \langle r, r \rangle$ if $d^2 < eps$ then stop $v = r + (d/c)v$ $c = d$ output k, x, r end	function [output] = cg(input) $r = b - A^*x$ $v = r$ $c = r^*r$ for $k = 1 : M$ if $\text{sqrt}(v^*v) < delta$ break end $z = A^*z$ $t = c / \langle v^*, z \rangle$ $x = x + t^*v$ $r = r - t^*z$ $d = r^*r$ if $d.^2 < eps$ break end $v = r + (d/v)^*v$ $c = d$ end

In the first line of the Program 2 replace output by x, k, r and input by $a, b, x^{(0)}, M, \epsilon, eps, delta$.

Naturally, these prospective learnings can be modified and/or adapted.

Learning Activities

Each Learning Activities suggested can be implemented, adapted to a special reality, or substituted by others that are considered more pertinent.

1. Given the system $\mathbf{Ax} = \mathbf{b}$, $n = 2, 3, \dots$ the student should verify: if \mathbf{A} is symmetric and positive definite. It should also build explicitly the form quadratic $q(x) = \langle x, \mathbf{Ax} \rangle - 2\langle x, \mathbf{b} \rangle$, to obtain the point which possesses their minimum value, and finally to compare this point with the exact solution of the system $\mathbf{Ax} = \mathbf{b}$, $n = 2, 3, \dots$

Example: The exact solution of LS: $\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$ is given by $x_1 = -6/5, x_2 = 7/5$, the following instructions in MAPLE (Worksheets 1) they allow to generate the quadratic form $q(x) = \langle x, \mathbf{Ax} \rangle - 2\langle x, \mathbf{b} \rangle$ and to obtain its minimum value.

Worksheets 1

```
> with (linalg):
> A:=matrix (2, 2, [2, 1, 1, 3]);
> b:=matrix (2, 1, [x1, x2]);
> x:=matrix (2, 1, [x1, x2]);
> q(x):=simplify(evalm(transpose(x) & A &
  *x-2* transpose(x) & *b));
> readlib(extrema):
> extrema(q(x),{x}, {u,v},'s');
> with (plots):
> plot3d(q(x),x1=-5..5, x2=0..5);
> minimize(q(x));
```

Another system on which you could apply a similar worksheets to the Worksheets 1 is formed for

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & -1 \\ -2 & -10 & 0 \\ -1 & -1 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ -12 \\ 2 \end{bmatrix}$$

Notice that the exact solution is given for $x_1 = x_2 = x_3 = 1$, while the application of the Worksheets 1 to this system hurls as a result that the minimum value of $q(x)$, it is reached in $x_1 = 3/2, x_2 = 1, x_3 = 1$. However, notice that the main matrix, \mathbf{A} , is not symmetrical and it is not positive definite, in fact, it possesses a negative eigenvalue. This activity for the learning allows the student to value the importance of the hypotheses in the mathematical theorems.

2. The systems idealized mass-spring as having numerous applications in the whole engineering. One has an arrangement of 4 springs in series that are compressed with a force F . In the balance, the

following LS of forces can be developed defining the interrelations among the springs:

$$\begin{bmatrix} -k_1 - k_2 & k_2 & 0 & 0 \\ k_2 & -k_2 - k_3 & k_3 & 0 \\ 0 & k_3 & -k_3 - k_4 & k_4 \\ 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ F \end{bmatrix}$$

The following Worksheets 2 of MAPLE allow to solve the system in exact form:

Worksheets 2

```
> eq[1]:=k[2]*(x[2]-x[1])=k[1]*x[1];
> eq[2]:=k[3]*(x[3]-x[2])=k[2]*(x[2]-x[1]);
> eq[3]:=k[4]*(x[4]-x[3])=k[3]*(x[3]-x[2]);
> eq[4]:=F=k[4]*(x[4]-x[3]);
> solve({eq[1], eq[2], eq[3], eq[4]}, {x[1], x[2],
  x[3], x[4]});
```

Table 2 shows the exact solution obtained with MAPLE.

Notice that one of the advantages of having the exact solution resides in the possibility of varying the constants of the springs freely together with the value of the force F . Together with the above-mentioned specific values can be assigned to the constants and to solve the LS using a direct method, GS and/or CG.

3. Build a program in MATLAB for the iterative methods: GS and CG.

Example: See Programs 1 and 2 of this article, for the GS and CG methods, respectively. Notice the didactic value that possesses the correspondence among the Program 2 and the Algorithm 2, which arises as consequence of using MATLAB. This correspondence allows a soft transition among the mathematical foundations (Algorithm 1) and the computational implementation (Program 2).

4. Test the Programs 1 and 2 in LS generated by the numeric solution of Partial Differential Equations.

Example: Consider the problem that consist of finding a function $u(x, y)$, such that

$$\begin{aligned} \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} &= x + y, \text{ for all } (x, y) \in D \\ &= \{(x, y) \in \mathbb{R}^2 : 0 < x < 2 \\ &\quad \text{and } 1 < y < 2.5\}, \end{aligned}$$

Table 2 Exact Solution

χ_1	χ_2	χ_3	χ_4
$\frac{F}{k_1}$	$\frac{F(k_1 + k_2)}{k_1 k_2}$	$\frac{F(k_1 k_2 + k_2 k_3 + k_1 k_3)}{k_1 k_2 k_3}$	$\frac{F(k_1 k_2 k_3 + k_1 k_2 k_4 + k_2 k_3 k_4 + k_1 k_3 k_4)}{k_1 k_2 k_3 k_4}$

and $u(x, y) = xy$, for all (x, y) , in the boundary of D .

It is possible to obtain a discrete version of $\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = x + y$, if the approximation formulas $\frac{\partial^2 u(x, y)}{\partial x^2} \cong \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2}$, and $\frac{\partial^2 u(x, y)}{\partial y^2} \cong \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2}$, are used.

The evaluation of this discrete outline, with $h = 1/2$, it generates the LS, $A\tilde{x} = \tilde{b}$, $n = 6$, where,

$$A = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \text{ and } \tilde{b} = \begin{bmatrix} 1.875 \\ 3.25 \\ 8.625 \\ 1 \\ 1.625 \\ 5.25 \end{bmatrix}$$

A simple application of the command *solve* of MAPLE allows us to obtain, at least, 19 digits of

$$\text{the exact solution : } \tilde{x} = \begin{bmatrix} 1321/966 \\ 3233/1288 \\ 6667/1932 \\ 524/483 \\ 635/322 \\ 5155/1932 \end{bmatrix} = \begin{bmatrix} 1.3674948240165631470 \\ 2.510093167701863540 \\ 3.4508281573498964803 \\ 1.0848861283643892340 \\ 1.9720496894409937888 \\ 2.6682194616977225673 \end{bmatrix}.$$

The application of the Program 1 (Gauss–Seidel) generates the integer and 15 decimals (format *long* of MATLAB) of the exact solution after 40 iterations, while the Program 2 (Conjugated Gradient) generates the same thing in 6 iterations. Notice that the 6 iterations required by the CG method illustrate the result fundamental content in the Fundamental Theorem.

Similarly, if $h = 1/10$, GS needs 301 iterations while CG needs 95, where the order of LS is $n = 266$.

5. Make an empiric comparison among the methods GS and CG if the LS is not well conditioned.

Example: Solve the LS: $A\tilde{x} = \tilde{b}$, where $a_{ij} = \frac{1}{i+j+1}$, $b_i = \frac{1}{3} \sum_{j=1}^n a_{ij}$, $1 \leq i, j \leq n$, using the GS and CG methods. The exact solution is $x_i = 1/3, i = 1, \dots, n$. The Program 3 (*m-file* of MATLAB), generates the matrix **A** and the vector **b**, for all n .

Programa 3

function [A,b]=hilbert(n)

for i=1:n;

for j=1:n;

A(i,j)=1/(i+j+1);

end

end

for i=1:n;

b(i)=(1/3)*sum(A(i,:));

end

b=b';

The Sequence 1, implemented in the line of commands of MATLAB, solves the LS for $n = 20$, using GS and CG methods, and graph both solutions:

Sequence 1

> [A,b]=hilbert(20);

> x0=zeros(20,1); M=500; eps=1e-30; delta=1e-30;

> [x, k, r]=cg(A, b, x0, M, eps, delta);

> y=gs(A, b); t=1:20;

> plot(t, x, 'o', t, y, '--')

It is requested the student that interprets the obtained graph.

6. Consider the problem of electric engineering that consists on determining the electrostatic potential $V = V(x, y)$, such that, $-(\frac{\partial^2 V(x, y)}{\partial x^2} + \frac{\partial^2 V(x, y)}{\partial y^2}) = \frac{\rho}{\epsilon_{\text{psilon}}}$, on the annular region $D = \{(x, y) \in \mathbb{R}^2 : /x^2 + y^2 < 1 \wedge x^2 + y^2 > 1/3\}$, $y V(x, y) = g(x, y)$, along the interior and external boundary of D , where, *epsilon* is the coefficient of dielectricity and *rho* is the space charge density.

The command: **spy(numgrid('A', 50))**, of MATLAB, generates Figure 1, which shows the annular region with the points in which will approach the electrostatic potential

Suppose that *rho*, *epsilon* and $g(x, y)$, are such that, $\tilde{b} = (b_i), b_i = 1, \forall i = 1 : 1252$ for the LS $A\tilde{x} = \tilde{b}$ where, **A** is the matrix of a five-point finite difference approximation of Laplace's equation on the annular region D . The command of MATLAB: **A=delsq(numgrid('A',50))** it generates **A**. The command **spy(A)** generates Figure 2, i.e., sparsity patron of **A**. The order of **A** is $n = 1252$.

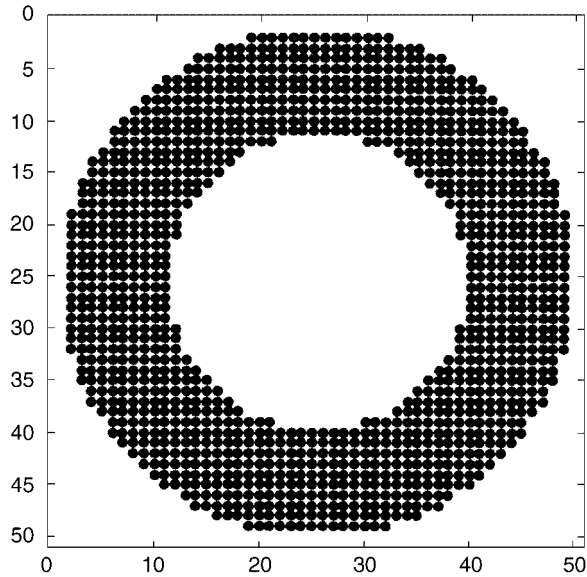


Figure 1 Grid on the annular region.

The sequence 2, implemented in the line of commands of MATLAB, solves the LS, using CG, methods:

Sequence 2

```
> A = delsq (numg rid('A', 50)); b = ones(size
(A, 1),1);
> x0 = zeros (size (b)) ; M = 500; eps = 1e-30;
delta = 1e-30;
> [x, k, r] = cg (A, b, x0, M, eps, delta);
```

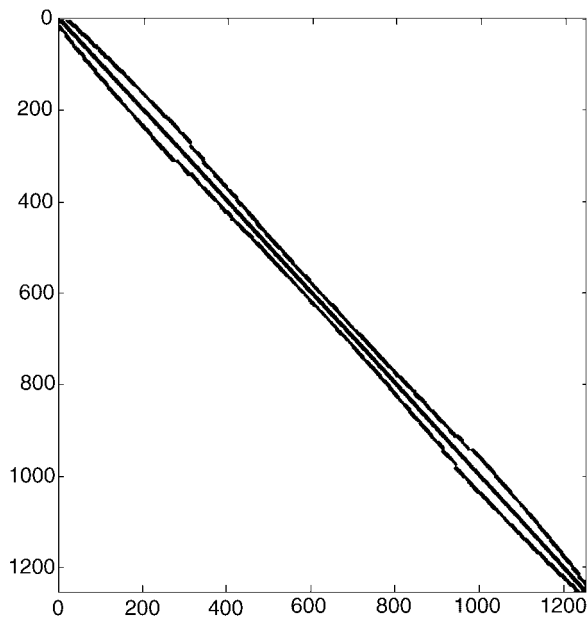


Figure 2 Sparsity patron. Matrix of principal coefficient.

Activities of Evaluation

The evaluation is considered part of the learning process. It should provide the student and the professor of the necessary feedback as relating to continue, to correct, and to guide and future activities.

CONCLUSIONS

In this article, we have considered a wide spectrum of learning experiences, supported by computational tools of last generation, in the environment of the course NA, belonging to most of the engineering curriculum. We have demonstrated how the combined use of MAPLE and MATLAB strengthens the process of teaching-learning, also allowing the learning of contents that usually are not treated at undergraduate level, as for example, Conjugates Gradient Methods. A close examination of our article and implementation reveals that alternative solutions in conventional programming language such as Fortran or C would require considerable effort. Most of the devices here have originated from courses given by the authors. MAPLE and MATLAB code, written by the authors, were provided to the students in the form of hardcopy. We plan to apply the methodology proposed to other areas of the NA.

REFERENCES

- [1] E. A. Cariaga, Combination of MATLAB 5.3 and MAPLE V in the teaching of the numerical analysis for engineering, Proceedings of the XIII Chilean Congress on Engineering Education, 1999, p 37.
- [2] E. A. Cariaga, Methods of the type conjugated gradient and their incorporation to the course of numerical analysis for engineering, Proceedings of the XIV Chilean Congress on Engineering Education, 2000, p 45.
- [3] ABET (Accreditation Board for Engineering and Technology), ABET Engineering Criteria 2000, Report of Engineering Accreditation Commission Criteria Committee (1995), 3 pages.
- [4] W. M. Philips, The Opportunity for Engineering Education and Accreditation in a Global Economy, Proceedings of the XIII Chilean Congress on Engineering Education (1999), p 14.
- [5] SIAM, "The SIAM Report on Mathematics in Industry," SIAM, 1998.
- [6] Ll. N. Trefethen, Predictions for Scientific Computing Fifty Years from Now, Oxford University Computing Laboratory, 1998.

- [7] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, 1997.
- [8] L. N. Trefthen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997, p 321.
- [9] S. C. Chapra and R. P. Canale, *Numerical methods for Engineer with Programming and Software Applications*, 3rd. edn., McGraw-Hill Companies, Inc. (1999), pp 235–349.
- [10] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 1st edn., Brooks/Cole Publishing Company, Wadsworth Inc., Belmont, California (1991), pp 185–223.

BIOGRAPHIES



Emilio Cariaga is an assistant professor in the Department of Mathematical and Physical Sciences at the Catholic University of Temuco, Chile. He was born in Valdivia, Chile, 1967. He received BS (1992) and MS (1996) degrees in mathematics from the Universidad Técnica Federico Santa María, Valparaíso, Chile. His interests are in computer applications in engineering education and interest in modeling of transport phenomena.

M. C. Nualart is an assistant professor in the School of Informatics at the Catholic University of Temuco, Chile. She received BS degree (1992) in Informatics Engineering from the University of Concepción, Chile. Her interests are in computer applications in engineering education and in software development.