

NUMERICAL SOLUTION FOR FREDHOLM
FIRST KIND INTEGRAL EQUATIONS OCCURRING
IN SYNTHESIS OF ELECTROMAGNETIC FIELDS*

ELENA BĂUTU, ELENA PELICAN

“Ovidius” University, Constanta, 900527, Romania
{crogojina, epelican}@univ-ovidius.ro

Received September 12, 2006

It is known that Fredholm integral equations of the first kind

$$\int_0^1 k(s, t)x(t)dt = y(s), \quad s \in [0, 1] \quad (1)$$

with the kernel $\frac{(s-t)^m}{[1-(s-t)^2]^n}$ occur when solving with problems of synthesis of electrostatic and magnetic fields (m, n – nonnegative rational numbers). This paper presents two approaches for solving such an equation. The first one involves discretization by a collocation method and numerical solution using an approximate orthogonalization algorithm. The second method is based on a nature inspired heuristic, namely genetic programming. It applies genetically-inspired operators to populations of potential solutions in the form of program trees, in an iterative fashion, creating new populations while searching for an optimal or near-optimal solution to the problem at hand. Results obtained in experiments are presented for both approaches.

Keywords: Fredholm integral equations of the first kind, genetic programming, inverse problems.

1. INTRODUCTION

Problems in the classical theory of electromagnetic fields come from two fundamental groups: analysis and synthesis of fields. Most papers in the literature deal with analysis of certain fields – finding the field distribution when we know the properties and geometry of the environment.

In this paper, we will consider the inverse problem – given the field distribution, we want to determine the cause that generated the assumed field. In the literature, this is known as the *synthesis of the field* problem. The problem

* Paper presented at the 7th International Balkan Workshop on Applied Physics, 5–7 July 2006, Constanta, Romania.

may have no solution, or it may have many different solutions; there is no universal algorithm for the synthesis of fields. The problem of generating fields with a given distribution has many implications in many branches of applied physics and technology. This paper tackles synthesis of electrostatic field on the symmetry axis of an axially symmetric system by reducing the problem to solving a Fredholm integral equation of the first kind.

2. THE SYNTHESIS OF FIELDS PROBLEM

The problem of synthesis of an electrostatic field considered in this paper is obtained from the following models. First, it is considered an electrode in the form of an elementary ring of length da and radius R , charged with $2\pi Rqda$. The potential dV and the component dE_z of the electrical field intensity on axis are expressed by the dependencies:

$$dV = \frac{qRda}{2\epsilon\sqrt{[R^2 + (z-a)^2]}}, \quad dE_z = \frac{Rq}{2\epsilon} \frac{z-a}{\sqrt{[R^2 + (z-a)^2]}^3} \quad (2)$$

The dimensionality of the problem is a very important issue, especially for the heuristic approach described in section 4. In order to transform the problem into dimensionless form, the following notations are considered:

$$x = \frac{z}{R}, \quad s = \frac{a}{R}, \quad s_0 = \frac{1}{R}$$

$$v(x) = V(x) \frac{2\epsilon}{R}, \quad e(x) = 2\epsilon E(x)$$

The resulted formula for E_z and V are given by the following Fredholm integral equations of the first kind:

$$v(x) = \int_{s_0}^{s_0} \frac{q(s)ds}{\sqrt{[1 + (x-s)^2]}}, \quad e(x) = \int_{-s_0}^{s_0} \frac{(x-s)q(s)ds}{\sqrt{[1 + (x-s)^2]}^3} \quad (3)$$

The second model involves a planar system, with electrodes supplied symmetrically (in which case component E_x of the vector E will occur on axis x) or anti-symmetrically in respect of axis y (component E_y will occur). Synthesis of the field in the planar system in the case of symmetric supply is described by the following equation

$$e_x(x) = \int_{-s_0}^{s_0} \frac{\tau(s)(x-s)}{1 + (x-s)^2} ds \quad (4)$$

and for anti-symmetric supply

$$e_y(x) = \int_{-s_0}^{s_0} \frac{\tau(s)}{1+(x-s)^2} ds \quad (5)$$

The two modern computational methods for treating first kind integral equations described in sections 3 and 4 will be verified upon an example of synthesis of the electrostatic field in the planar anti-symmetrical system described by equation (5) with $e(x) = 1$, and $s_0 = 1$.

3. NUMERICAL TREATMENT

Fredholm integral equations of the first kind like (3) are a special case of inverse problems, severely ill-posed. In the case of ill-posed problems, the main problem is the instability of the solution: small changes in the right hand side result in big perturbation of the solution. Input data usually comes from measurements, and measurements are prone to error; hence, the instability of the solution is a very important issue. Classical treatment of such problems requires careful use of regularization methods or other special numerical algorithms.

The first step in the numerical treatment used in this research consists in discretization of equation (3) by a collocation method [11]. The collocation method that was used is a projection method in which the approximate solution is determined from the condition that the equation be satisfied at certain given points, called collocation nodes. For $n \geq 2$ arbitrary fixed and $T_n = \{t_1 \dots t_n\}$ the set of collocation points in $[0, 1]$ ($0 \leq t_1 < t_0 < \dots < t_n = 1$), we consider the collocation discretization of (1). The problem becomes: find $x \in L^2([0, 1])$ such that

$$\int_0^1 k(s, t_i)x(t_i) = y(s), \quad s \in [0, 1] \quad (6)$$

For $t_i \in T_n$ we define $k_{t_i} : [0, 1] \rightarrow \mathbb{R}$ by

$$k_{t_i}(s) = k(t_i, s), \quad \forall s \in [0, 1] \quad (7)$$

In [11] there is proof that the minimal norm solution in the least squares sense for equation (1) can be computed by

$$x_n^{LS}(t) = \sum_{j=1}^n \alpha_j k(s_j, t), \quad t \in [0, 1] \quad (8)$$

where $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ is the minimal norm solution of the system $A_n \alpha = b_n$. The elements for matrix A_n and vector b_n are given by

$$(A_n)_{ij} = \int_0^1 k(s_i, t)k(s_j, t)dt, \quad (b_n)_i = y(s_i), \quad i, j = 1, \dots, n \quad (9)$$

The ill-posed nature of the problem leads to an ill-conditioned linear system. Unlike the case of well-posed problems, in this case, refining the discretisation will lead to a very ill-conditioned system matrix. It is also rank-deficient, and symmetric – due to the nature of the discretisation technique by means of collocation method. Direct methods or classical iterative schemes are useless for this type of system. A family of iterative solvers for relatively dense symmetric linear systems based upon Kovarik’s approximate orthogonalization algorithm is considered in this paper. Although the original method developed in [1] was designed for a set of linearly independent vectors in a Hilbert space, it was recently extended to an arbitrary set of vectors in \mathbb{R}^n in [2]. We used the adaptation for symmetric matrices available in [3]. Thus, the algorithm used for the linear system is K OBS-rhs (see [4] for details).

K OBS-rhs Algorithm Let $A_0 = A$, $b^0 = b$. For $k = 0, 1, \dots$ do

$$K_k = (I - A_k)S(A_k; n_k), \quad A_{k+1} = (I + K_k)A_k, \quad b^{k+1} = (I + K_k)b^k$$

4. GENETIC PROGRAMMING

Evolutionary Algorithms are adaptive algorithms based on the Darwinian principle of natural selection. Genetic programming (GP) is an automated method from the class of EAs designed for the task of automatically creating a computer program that solves a given problem. GP accomplishes this by breeding a population of computer programs over many generations using the operations of selection, crossover (sexual recombination), and mutation. Since it was proposed by Koza in [5], it was successfully applied to problems in various fields of research, such as pattern recognition, data mining, robotic control, bioinformatics and even picture and music generation. The algorithm used in this paper is Gene Expression Programming (GEP), a revolutionary methodology based on GP proposed by Ferreira in [10] for discovering knowledge from data, in the form of the symbolic expression of a function.

One key issue in the inverse problem tackled in this paper is the choice of the search space, *i.e.*, in Evolutionary Computation terminology, of the representation. Since GP works with program trees, it is easily applied to finding approximate solutions to first kind integral equations in the form of complex compositions of functions. In classic GP, the chromosomes are hierarchically organized structures (syntax trees) of different sizes and shapes. This feature makes genetic operators rather difficult to implement and lowers the degree of efficiency in applying them. GEP individuals – also called chromosomes – are linear expressions of fixed length composed of functions and terminals, which

encode expression trees of various structures. The fixed length structure of the individuals is a feature that GEP has inherited from classical genetic algorithms (GA). This linear representation leads to easy manipulation of the individuals through genetic operators that resemble very much the standard genetic operators encountered in the field of GAs.

On the other hand, the expression trees encoded by the chromosomes are very complex compositions of functions and terminals, and exhibit complex functional behavior. The main advantage over classical GP representation is that GEP representation offers separation between phenotype and genotype. The definition of a GEP gene allows modification of the genome using any genetic operator without restrictions, producing always syntactically correct programs.

4.1. REPRESENTATION

The chromosomes in GEP are multi-genic linear symbolic strings of fixed length. Each gene is composed of head and tail sections. The head may contain both functions and terminals, whereas the tail is limited to contain only terminals. Depending on the problem to be solved, the number of genes in the chromosome is a fixed parameter for a run. Also, the length of the head h is fixed, whereas the length of the tail t is a function of h and the maximum arity of the functions in the function set of the algorithm max_arity :

$$t = h(\text{max_arity} - 1) + 1$$

The values of the parameters of a GEP run are not fixed, but there exist empirical studies [6] that enable us to make educated guesses in choosing the appropriate values for them. For example, if $h = 5$, the set of functions is $F = \{\times, /, -, +\}$ and the set of terminals is $T = \{x, y\} \cup \mathbb{R}$, then $n = 2$ and $t = 5 \times (2 - 1) + 1 = 6$; thus, the length of the gene is $5 + 6 = 11$. One such gene, with the sequence of symbols that actively participate in the decoding of the gene shown underlined (this is called an Open Reading Frame in biology), might look like this

$$\underline{+ \times 4} \underline{x - y} \underline{3} \underline{x} \underline{5} \underline{y} \underline{2} \tag{10}$$

To decode the gene into an expression we need to consider the arity of each symbol (variables and numerical constants are considered 0-arity), and associate each symbol with the parameters necessary for its evaluation. For the previous gene, the first symbol (+) requires two parameters (the second and the third symbols, \times and 4); the second symbol (\times) requires two parameters (\times and $-$); the third and fourth symbols (4 and \times) require no parameters, and so on. Therefore, the gene decodes to the following expression: $((x) \times ((y) - (3))) + (4)$. When decoding a chromosome, the subexpressions encoded by the genes are linked by means of a linking function – the same for all chromosomes in the algorithm – usually addition.

4.2. FITNESS FUNCTION

Every individual in the genetic algorithm population is assigned a fitness value according to how well it solves the problem. Our symbolic approach to solving Fredholm integral equations of the first kind automatically induces a solution for (1) in symbolic form by means of the evolutionary paradigm described so far. In order to compute the fitness of a chromosome c (see [8]), we use Simpson's quadrature rule to approximate the integral in the left hand side of (1)

$$\int_0^1 k(s, t)x_c(t)dt \approx \sum_{i=0}^n \omega_i k(s, t_i)x_c(t_i) = \Phi_c(s)$$

where x_c is the expression (function) encoded by chromosome c and the weights are:

$$\omega_i = \begin{cases} 1/3n, & i = 0 \text{ or } i = n \\ 4/3n, & 0 < i < n \text{ and } i \text{ is odd} \\ 2/3n, & 0 < i < n \text{ and } i \text{ is even} \end{cases}$$

Further, we can now use collocation and compute the absolute error as the sum of the absolute error over the set of fitness cases

$$error_c = \sum_{s \in S} |\Phi_c(s) - f(s)| \quad (11)$$

where S is the set of collocation points. In our experiments, the set of collocation points is the same as the set of quadrature points; they are equidistant points in the interval $[0, 1]$, with the number of points being set at the beginning of a run (we used usually up to 50 points). In fact, this approach consists in solving the problem of symbolic regression [5, 9] with the data set obtained by collocation as described in this paper. The set of fitness cases for the symbolic regression formulation of the problem consists in the set of pairs, each pair composed of a collocation point and the value of the left hand side f of the equation in that point.

The selection scheme used in the experiments is roulette wheel selection combined with elitist survival of the best individual from one generation to the next. Every individual has fitness proportionate chances to survive and participate to genetic operations, and the best individual in a generation is sure to have at least a copy in the next generation.

4.3. GENETIC OPERATORS

A gene expression algorithm may use a large set of evolutionary operators to evolve the population: mutation, crossover, transposition, etc. According to

[6], mutation is the most efficient among the genetic operators when it comes to modification of the genome. Other studies in the field of GP emphasize the negative impact mutation may have on the convergence of the algorithm. Hence, we used a rather low rate for the mutation operator in our experiments. In GEP, mutations can occur anywhere in the chromosome as long as the structural organization of chromosomes is preserved. In the head, any symbol can change into any other symbol (function or terminal); in the tail, terminals can only change into terminals. The mutation rate in a GEP algorithm is much higher than the mutation rates found in nature (see [7]), but thanks to elitism, populations undergoing excessive mutation evolve very efficiently. For example, the gene from (10) may mutate into

$$\underline{+ y 4} x - y 3 x 5 y 2$$

which translates into the expression $(y) + (4)$.

Other GEP unary operators are the transposition operators: insertion sequence (IS) transposition, root IS transposition, and gene transposition. These operators copy a sequence of symbols called transposon (or insertion sequence) and insert it into some location. Transposition operators differ in the way they choose the insertion sequence and the insertion point. IS transposition chooses a random insertion sequence and inserts it at a random location in the head of the gene, except for the first position. For example, the gene from (10) may evolve into

$$\underline{+ 4 x} \times 4 y 3 x 5 y 2$$

which translates into the expression $(4) + (x)$. RIS transposition chooses a random insertion sequence that begins with a function and inserts it at the first position of the head of the gene. The gene from (10) may evolve into

$$4 + \times 4 b 3 a 5 b 2$$

which translates into the expression $(4) \times (((4) \times (b)) + (3))$. Gene transposition moves an entire gene in front of the first gene of the chromosome. This operator has no real effect in the case of a commutative linking function.

GEP also uses binary operators such as crossover that work by swapping information between two chromosomes: one-point crossover, two-point crossover and gene crossover. These operators work on two chromosomes by swapping information between two (or more) cutting points. The main difference between the operators afore mentioned lies in the way that cutting points are selected. The first cutting point in one-point crossover is the beginning of the chromosome; the second cutting point is randomly selected. The two-points crossover operator has both cutting points randomly selected. Gene crossover uses the first cutting point at the beginning of the chromosome and the second cutting point is located after some randomly selected gene.

5. EXPERIMENTS

The experiments were conducted on the problem of synthesis of the electrostatic field in the planar anti-symmetrical system in (5), with $e(x) = 1$ (synthesis of the homogeneous field), and $s_0 = 1$ using the two radically different approaches described in the previous sections. The performances of the two methods are then compared using as criterion the sum of the absolute errors defined in the manner described by equation (11).

Numerical experiments were carried out with 50 collocation points and 25 iterations of the MKOBS algorithm. The numerical solution obtained is depicted in Fig. 1. Experiments with 10, respectively 200 collocation points yielded results similar in the form to the solution in the 50 points case. This solution is the minimal norm solution in the least squares sense for the problem.

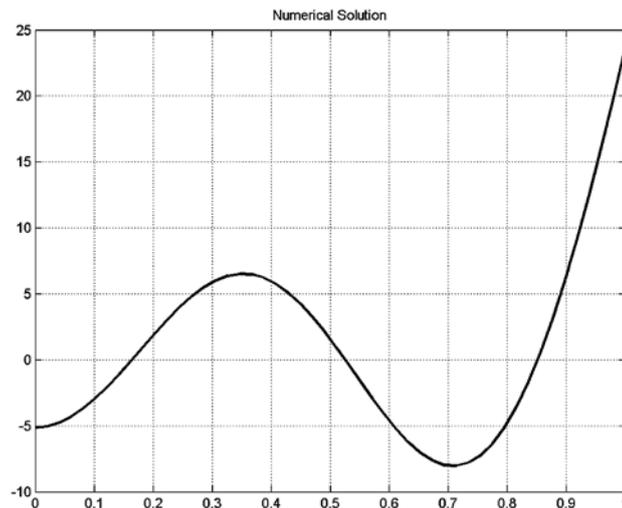


Fig. 1 – Solution obtained by means of collocation discretization followed by MKOBS algorithm.

GEP results were obtained in 50 separate runs, with fixed run parameters that can be consulted in Table 1. As it can be seen, the genetic algorithm searched in the space of trigenic chromosomes. After several experiments, we decided that a population size of 100 individuals and a number of 300 generations (which is quite small) is sufficient for the algorithm to find a satisfactory solution. The number of collocation points (and at the same time quadrature points) is the same as in the numerical experiments – 50.

The function set used includes mathematical operators such as addition, multiplication, protected division, and subtraction, exponential and protected logarithmic functions. Protected variants of functions return the value 1 whenever the result of the evaluation of the function is not computable. Experiments that included trigonometric functions in the function set did not

Table 1

Gene expression algorithm settings

Parameter	Value
Population size	100
Iterations	300
Number of genes	3
Gene size	9 or 13
Head size	6
Tail size	7 or 13
Linking function	addition (+)
Mutation rate	5%
Crossover rate	60%
Gene crossover rate	30%
Insertion sequence size	5
IS Transposition rate	10%
RIS Transposition rate	10%
Gene Transposition rate	0%
Random constant range	[-10, +10]
Function rate	50%
Constant rate	30%
Variable rate	20%

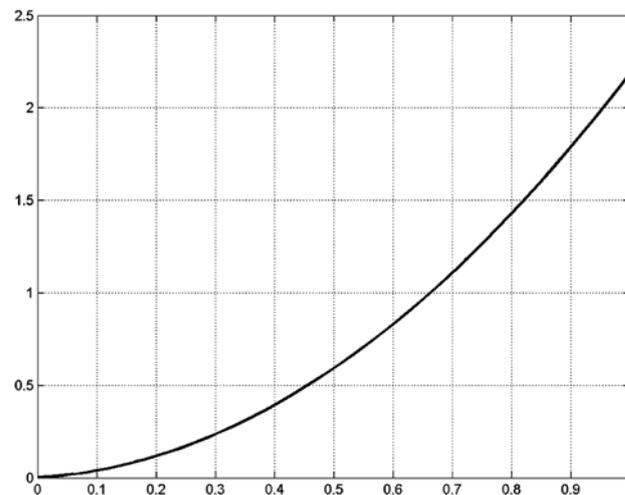


Fig. 2 – Solution obtained by GEP.

result in better solutions. The tail size is either 7 or 13 depending on whether the function set includes the ternary function *if* or not.

The solution obtained by GEP (see Fig. 2) is the best individual obtained in 50 different runs. This chromosome decodes to the actual solution in symbolic form, which has an error of 0.89 and looks quite simple:

$$x(s) = (((t/t) \times (t \times t)) + ((t \times t) + (t \times e^{-1.695301})))$$

A comparison of the solutions obtained based on the absolute error of the solution obtained with respect to the exact free term (see (11)) points out with clarity that the solution obtained by GEP is more fit, according to the criterion considered, to the problem we tackled in this paper.

The deviation of the real field distribution obtained as a result of solving the synthesis problem from the distribution assumed on axis for GEP solutions:

$$d(s) = e_y(s) - 1.0.$$

It may easily be noted that the difference obtained by the GEP formula is closer to 0 than the difference of the solution of the collocation followed by MKOBS approach. In the case of the integral equation with the right hand side

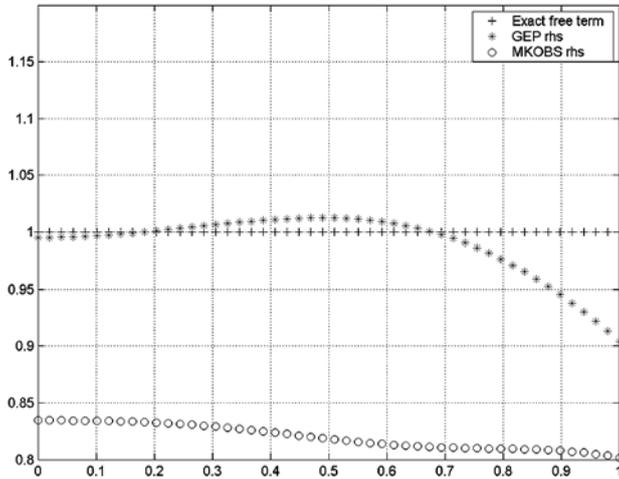


Fig. 3 – Absolute error between the integral approximation of the rhs and the exact value of the lhs $e_y(s) = 1$.

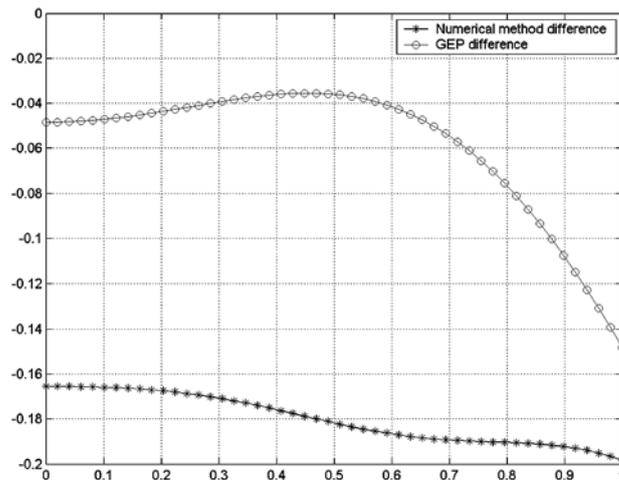


Fig. 4 – Deviation of the real field distribution obtained as a result of solving the synthesis problem with the numerical method described, and GEP respectively.

$y(s) = \text{atan}(s+1) - \text{atan}(s-1)$ (computed such that the exact solution to the equation is the constant function $x(t)=1$), the numerical method based on collocation and KOBS computes the solution with $\|d(s)\|_{\text{inf}} = 6.7e-3$.

The results presented above operate with electric charge that should be accumulated on the electrodes. In practice, charge is a quantity difficult to measure and regulate precisely. Electric potential is easier to realize on the electrodes. It is easy to determine the potential (see [12] for details).

6. CONCLUSIONS

At least one cause must exist for any given effect. Therefore, the existence of a solution for an inverse problem that models a physical system is usually not an issue. The ill-posed character of inverse problems with physical interpretation is usually due to the impossibility to demonstrate the uniqueness of a solution (since many times, there may exist different causes that determine similar effects) and to the instability of such a solution. The numerical approach described in this paper offers a method of discretization and suggests an algorithm for solving the resulted linear system. Thus, after applying formula (8) we obtain the minimal norm solution in the least squares sense for the initial integral equation.

On the other hand, the heuristic approach by means of genetic programming has the advantage that it provides us with distinct, alternative equally fit solutions for the same problem according to the criterion of minimizing the error specified by (11). In the context of searching for an unknown cause of a given effect, this approach could offer insights and promising models that can be further analyzed and developed by specialists in the field of electromagnetism.

Acknowledgements. This paper was supported by the CNMP CEEX-05-D11-25/2005 Grant.

REFERENCES

1. S. Kovarik, *Some iterative methods for improving orthogonality*, SIAM J. Num. Anal. 7 (1970), 386–389.
2. C. Popa, *A method for improving orthogonality of rows and columns of matrices*, Intern.J. of Comp. Math., 77 (2001), 469–480.
3. C. Popa, *On a modified Kovarik algorithm for symmetric matrices*, Annals of “Ovidius” Univ., Series Mathematics, vol XI(1)2003, 147–156.
4. M. Mohr, C. Popa, U. Rude, *A Kovarik Type Algorithm without Matrix Inversion for the Numerical Solution of Symmetric Least-Squares Problems*, Lehrstulbericht 05-2, 2005.
5. J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.

6. C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*, Angra do Heroísmo, Portugal, 2002, On-line: <http://www.gene-expression-programming.com/gep/Books/>.
7. D. J. Futuyma, *Evolutionary Biology*, 3rd ed., Sunderland, MA: Sinauer Associates, 1998.
8. E. Băutu, A. Băutu, H. Luchian, *A GEP-based approach for solving Fredholm first kind integral equations*, Proceedings of SYNASC 2005, IEEE Computer Society Press, Los Alamitos, CA, USA, ISBN 0-7695-2453-2.
9. E. Băutu, A. Băutu, H. Luchian, *Symbolic Regression on Noisy Data with Genetic and Gene Expression Programming*, Proceedings of SYNASC 2005, IEEE Computer Society Press, Los Alamitos, CA, USA, ISBN 0-7695-2453-2.
10. C. Ferreira, *Gene Expression Programming: A New Adaptive Algorithm for Solving Problems*, Complex Systems, 2001.
11. E. Pelican, *Extensions of Projections and Collocation Methods for First Kind Integral Equations*, Revue Roumaine de Mathématiques Pures et Appliquées, 3(51), 2006, to appear.
12. K. Adamiak, *On Fredholm Integral Equations of the First Kind Occurring in Synthesis of Electromagnetic Fields*, Intern. J. for Numerical Methods in Engineering, Vol. 17, 1187–1200 (1981).