Proceedings of the 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING, Lisbon, Portugal, June 16-18, 2005 (pp36-42)

Numerical Solution of Non-Linear Ordinary Differential Equations via Collocation Method (Finite Elements) and Genetic Algorithms

Nikos E. Mastorakis
Military Institutes of University Education (ASEI)
Hellenic Naval Academy, Terma Hatzikyriakou 18539,
Piraeus, GREECE

http://www.wseas.org/mastorakis

Abstract: - In this paper a new method for solving (non-linear) ordinary differential equations is proposed. The method is based on finite elements (collocation method) as well as on genetic algorithms. The method seems to have some advantages in comparison with the typical sequential (one – step and multi – step) methods.

Key-Words: - Ordinary Differential Equations, Finite Elements, Genetic Algorithms, Evolutionary Computing, Collocation

1 Introduction

Research in numerical solution of Ordinary Differential Equations (ODEs) is an open field during the last centuries and many numerical methods have been adopted to solve initial value problems. The importance of ODE is great because mathematical models that occur in science (physics, chemistry, biology, economy, geology, space science, etc) and engineering (electrical, mechanical, civil, etc…) are usually ordinary differential equations.

Euler's method, Heun's method, Polygon method, Runge–Kutta methods are the most usual one–step methods, while Milne–Simpson, Adams–Bashforth–Moulton methods are the most usual multi–step method. Suppose that a first–order ODE is expressed in the form: $\frac{dy}{dx} = f(x, y)$

Common feature in one–step methods is the "discovery" and the usage of a (discrete) sequence $y_{i+1} = y_i + F(x_i, y_i)$ where F is a function or procedure or algorithm in which x_i and y_i are involved.

Common feature in the so–called multistep methods is the "discovery" and the usage of a (discrete) sequence

$$y_{i+1} = y_i + F(x_i, y_i, x_{i-1}, y_{i-1}, x_{i-2}, y_{i-2}, \cdots, x_{i-k}, y_{i-k})$$

The previous formula describes a k – step method.

More details can be found in many textbooks and surveys like [1] ÷ [3].

A disadvantage of all these methods is that the numerical errors are accumulated and so the solution of the initial value problems is affected by these errors.

Especially when the interval of solution is big, we have to use small step of increase which affects the complexity of the method. Therefore, to increase accuracy we must decrease the step of the method, which unavoidably leads to big computational load and complexity.

In this paper, an attempt is made to solve this problem by using finite elements (collocation method) and genetic algorithms. Finite elements' method yields a set of non–linear algebraic equations. These non–linear equations can be solved via Genetic Algorithm (GA) [4]. The solution of the non–linear (algebraic) equations via GAs is examined in [4] with more details and specific examples.

In section 2, we present the main "finite elements". In section 3, a formulation of the general initial value problem is given. In section 4, a formulation of the general boundary value problem is presented. Finally, some concluding remarks are provided in 5.

2 Overview of the Main Finite Elements

"Finite Elements" are bases of functions such that the solution of an initial (or boundary) value problem to be able to be expressed as a linear combination of these functions. Below we remind the most common finite elements. Finite elements are basis functions that are nonzero only in small regions ("the elements").

Suppose that we have to solve the following initial value problem over the interval $[x_0, x_{n+1}]$

$$\frac{dy}{dx} = f(x, y) \tag{1}$$

with the initial condition

$$y(x_0) = y_0 \tag{1.a}$$

2.1 Finite Elements: Piecewise Linear Functions ("hat functions")

In the interval $[x_0, x_{n+1}]$ we consider the points:

$$x_0 \langle x_1 \langle x_2 \langle \cdots \langle x_n \langle x_{n+1} \tag{3}$$

as well as the functions

$$\phi_0 = \begin{cases} \dfrac{x_1 - x}{x_1 - x_0} & x_0 \le x \le x_1 \\ 0 & elsewhere \end{cases} \tag{4.1}$$

$$\phi_j = \begin{cases} \dfrac{x - x_{j-1}}{x_j - x_{j-1}} & x_{j-1} \le x \le x_j \\ \dfrac{x_{j+1} - x}{x_{j+1} - x_j} & x_j \le x \le x_{j+1} \\ 0 & elsewhere \end{cases} \tag{4.j}$$

$$\phi_{n+1} = \begin{cases} \dfrac{x - x_n}{x_{n+1} - x_n} & x_n \le x \le x_{n+1} \\ 0 & elsewhere \end{cases} \tag{4.n+1}$$

It is proved that the functions ϕ_j compose a basis of the space of functions that are continuous in the interval $[x_0, x_{n+1}]$. In applications we consider the solution of our (Non-Linear) Ordinary Differential Equations as a linear combination of the finite subset $\{\phi_0, \phi_1 \cdots, \phi_n, \phi_{n+1}\}$ of this basis.

Usually the points $x_0, x_1, x_2, \cdots, x_n, x_{n+1}$ are equispaced. So

$$\phi_0 = \begin{cases} \dfrac{x_1 - x}{h} & x_0 \le x \le x_1 \\ 0 & elsewhere \end{cases} \tag{5.1}$$

$$\phi_j = \begin{cases} \dfrac{x - x_{j-1}}{h} & x_{j-1} \le x \le x_j \\ \dfrac{x_{j+1} - x}{h} & x_j \le x \le x_{j+1} \\ 0 & elsewhere \end{cases} \tag{5}$$

$$\phi_{n+1} = \begin{cases} \dfrac{x - x_n}{h} & x_n \le x \le x_{n+1} \\ 0 & elsewhere \end{cases} \tag{5.n+1}$$

2.2 Finite Elements: Piecewise Square Functions

In the interval $[x_0, x_{n+1}]$ we consider the points: $x_0 \langle x_1 \langle x_2 \langle \cdots \langle x_n \langle x_{n+1}$

We define $x_0^{'} = \dfrac{x_0 + x_1}{2}$, $x_1^{'} = \dfrac{x_1 + x_2}{2}, \ldots,$ $x_{j-1}^{'} = \dfrac{x_{j-1} + x_j}{2}$, $x_n^{'} = \dfrac{x_n + x_{n+1}}{2}$

We also consider

Proceedings of the 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING, Lisbon, Portugal, June 16-18, 2005 (pp36-42)

$$\phi_j = \begin{cases} \dfrac{(x - x'_{j-1}) \cdot (x - x_{j-1})}{(x_j - x'_{j-1}) \cdot (x_j - x_{j-1})} & x_{j-1} \le x \le x_j \\ \dfrac{(x - x'_j) \cdot (x - x_{j+1})}{(x - x'_j) \cdot (x_j - x_{j+1})} & x_j \le x \le x_{j+1} \\ 0 & elsewhere \end{cases} \quad (6)$$

and

$$\psi_j = \begin{cases} \dfrac{(x - x_j) \cdot (x - x_{j+1})}{(x'_j - x_j) \cdot (x'_j - x_{j+1})} & x_j \le x \le x_{j+1} \\ 0 & elsewhere \end{cases} \quad (7)$$

Especially

$$\phi_0 = \begin{cases} \dfrac{(x - x'_0) \cdot (x - x_1)}{(x_1 - x'_0) \cdot (x_0 - x_1)} & x_0 \le x \le x_1 \\ 0 & elsewhere \end{cases} \quad \text{and}$$

$$\phi_{n+1} = \begin{cases} \dfrac{(x - x'_n) \cdot (x - x_n)}{(x_{n+1} - x'_n) \cdot (x_{n+1} - x_n)} & x_n \le x \le x_{n+1} \\ 0 & elsewhere \end{cases}.$$

It is proved that the functions $\{\psi_0, \phi_0, \psi_1, \phi_1 \cdots, \psi_n, \phi_n, \psi_{n+1}, \phi_{n+1}, ...\}$ compose a basis of the space of functions that are continuous in the interval $[x_0, x_{n+1}]$. In applications, we consider the solution of our (Non-Linear) Ordinary Differential Equations as a linear combination of the finite subset $\{\psi_0, \phi_0, \psi_1, \phi_1 \cdots, \psi_n, \phi_n, \psi_{n+1}, \phi_{n+1}\}$ of this basis.

2.3 Finite Elements: Hermite Functions

The basic Hermite functions are defined as follows

$$\phi(x) = \begin{cases} (x-1)^2 \cdot (2x+1) & 0 \le x \le 1 \\ (x+1)^2 \cdot (-2x+1) & -1 \le x \le 0 \\ 0 & elsewhere \end{cases} \quad (8)$$

$$\psi(x) = \begin{cases} x(x-1)^2 & 0 \le x \le 1 \\ x(x+1)^2 & -1 \le x \le 0 \\ 0 & elsewhere \end{cases} \quad (9)$$

For $j = 1, 2, \cdots, n$ we define

$$\phi_j(x) = \begin{cases} \phi\left(\dfrac{x - x_j}{\frac{x_{j+1} - x_{j-1}}{2}} \right) & x_{j-1} \le x \le x_{j+1} \\ 0 & elsewhere \end{cases} \quad (10)$$

$$\psi_j(x) = \begin{cases} \dfrac{x_{j+1} - x_{j-1}}{2} \cdot \psi\left(\dfrac{x - x_j}{\frac{x_{j+1} - x_{j-1}}{2}} \right) & x_{j-1} \le x \le x_{j+1} \\ 0 & elsewhere \end{cases} \quad (11)$$

as well as

$$\phi_0(x) = \begin{cases} \phi\left(\dfrac{x - x_0}{\frac{x_1 - x_{-1}}{2}} \right) & x_0 \le x \le x_1 \\ 0 & elsewhere \end{cases} \quad (10.0)$$

$$\phi_{n+1}(x) = \begin{cases} \phi\left(\dfrac{x - x_{n+1}}{\frac{x_{n+2} - x_{n-1}}{2}} \right) & x_n \le x \le x_{n+1} \\ 0 & elsewhere \end{cases} \quad 10.n+1)$$

$$\psi_0(x) = \begin{cases} \dfrac{x_1 - x_{-1}}{2} \cdot \psi\left(\dfrac{x - x_0}{\frac{x_1 - x_{-1}}{2}} \right) & x_0 \le x \le x_1 \\ 0 & elsewhere \end{cases} \quad (11.0)$$

$$\psi_{n+1}(x)=\begin{Bmatrix} \frac{x_{n+2}-x_n}{2}\cdot\psi\left(\frac{x-x_{n+1}}{x_{n+2}-x_n}\right) & x_n\le x\le x_{n+1} \\ 0 \quad elsewhere & \end{Bmatrix} \quad (11.n)$$

We consider again the solution of the differential equation as a linear combination of the finite subset $\{\phi_0,\psi_0,\phi_1,\psi_1,\phi_2,\psi_2,\cdots,\phi_n,\psi_n,\phi_{n+1},\psi_{n+1}\}$ of this basis.

2.4 Finite Elements: Splines Functions

The basic procedure spline function is defined as follows

$$\phi(x)=\begin{Bmatrix} \frac{1}{4}(x+2)^3 & -2\le x\le -1 \\ \frac{1}{4}[1+3(x+1)+3(x+1)^2-3(x+1)^3] & -1\le x\le 0 \\ \frac{1}{4}[1+3(1-x)+3(1-x)^2-3(1-x)^3] & 0\le x\le 1 \\ \frac{1}{4}(2-x)^3 & 1\le x\le 2 \end{Bmatrix} \quad (12)$$

So the Spline basis is $\{\phi_0,\phi_1,\cdots,\phi_{n+1}\}$ where

$$\phi_j(x)=\begin{Bmatrix} \phi\left(\dfrac{x-x_j}{\frac{x_{j+2}-x_{j-2}}{4}}\right) & x_{j-2}\le x\le x_{j+2} \\ 0 \quad elsewhere & \end{Bmatrix} \quad (13)$$

$j=2,\cdots,n-1$.

Obviously a slight modification is needed for $\phi_0,\phi_1,\phi_n,\phi_{n+1}$

Hence $$\phi_0(x)=\begin{Bmatrix} \phi\left(\dfrac{x-x_0}{\frac{x_2-x_{-2}}{4}}\right) & x_0\le x\le x_2 \\ 0 \quad elsewhere & \end{Bmatrix} \quad (13.0)$$

$$\phi_1(x)=\begin{Bmatrix} \phi\left(\dfrac{x-x_1}{\frac{x_3-x_{-1}}{4}}\right) & x_0\le x\le x_3 \\ 0 \quad elsewhere & \end{Bmatrix} \quad (13.1)$$

$$\phi_n(x)=\begin{Bmatrix} \phi\left(\dfrac{x-x_n}{\frac{x_{n+2}-x_{n-2}}{4}}\right) & x_{n-2}\le x\le x_{n+1} \\ 0 \quad elsewhere & \end{Bmatrix}$$

$$\phi_{n+1}(x)=\begin{Bmatrix} \phi\left(\dfrac{x-x_{n+1}}{\frac{x_{n+3}-x_{n-1}}{4}}\right) & x_{n-1}\le x\le x_{n+1} \\ 0 \quad elsewhere & \end{Bmatrix} \quad (13.n+1)$$

We can consider again the solution of the differential equation as a linear combination of the finite subset $\{\phi_0,,\phi_1,,\phi_2,...,\phi_{n+1}\}$ of this basis.

3 Formulation and Solution of the General Initial Value Problem via Collocation and Genetic Algorithm

Consider the general first – order initial problem.

$$\frac{dy}{dx}=f(x,y) \quad (1)$$

$$y(x_0)=y_0 \quad (1.a)$$

Suppose that $\{\phi_j\}_{j=1}^n$ is a subset of the basis of functions (like § 2.1 or § 2.2 or § 2.3 or § 2.4), on which our solution is expanded. Then, our solution can be expressed as a linear combination

$$y(x)=\sum_{j=0}^{n}c_j\phi_j \quad (14)$$

In the method of collocation [11] we select n points in the interval $[x_0, x_{n+1}]$ $x_0 \langle x_1 \langle x_2 \langle \cdots \langle x_n \langle x_{n+1}$.

These points can be equispaced or non – equispaced.

Then we have the equations

$$y_0 = \sum_{j=0}^{n} c_j \phi_j (x_0) \qquad (15.0)$$

$$\sum_{j=0}^{n} c_j \phi_j^{'} \Big|_{x_1} = f\left(x_1, \sum_{j=0}^{n} c_j \phi_j (x_1) \right) \qquad (15.1)$$

$$\vdots \qquad\qquad \vdots$$

$$\sum_{j=0}^{n} c_j \phi_j^{'} \Big|_{x_n} = f\left(x_n, \sum_{j=0}^{n} c_j \phi_j (x_n) \right) \qquad (15.n)$$

We can solve the Equation (15.0) with respect to one c_j, say c_l, where $l \in \{0,1,...,n\}$. Substituting this c_l into (15.1)…(15.n), we obtain a system of *n* non–linear equations in *n* unknowns: $c_0, c_1,..., c_{l-1}, c_{l+1},..., c_n$. These equations are numbered as (15a.1)…(15a.n). Please, note that these equations do not contain c_l

$$\sum_{j=0}^{n} c_j \phi_j^{'} \Big|_{x_1} = f\left(x_1, \sum_{j=0}^{n} c_j \phi_j (x_1) \right) \qquad (15a.1)$$

$$\vdots \qquad\qquad \vdots$$

$$\sum_{j=0}^{n} c_j \phi_j^{'} \Big|_{x_n} = f\left(x_n, \sum_{j=0}^{n} c_j \phi_j (x_n) \right) \qquad (15a.n)$$

The solution of (15a.1), … , (15a.n) can be obtained via a Genetic Algorithm following the method of [6]. The Genetic Algorithm is used to eliminate an error in (15a.1), …, (15a.n). The reason that we solve (15.0) with respect to c_l, as well as the reason for executing the Genetic Algorithm on (15a.1),…,(15a.n) instead of (15.0),(15.1),…,(15.n) is that we must satisfy the initial condition (1.a) with zero error.

A brief overview of the GAs methodology could be the following: Suppose that we have to maximize (minimize) the function $Q(x)$ which is not necessary continuous or differentiable. GAs are search algorithms which initially were insiped by the process of natural genetics (reproduction of an original population, performance of crossover and mutation, selection of the best). The main idea for an optimization problem is to start our search no with one initial point, but with a population of initial points. The 2*n* numbers (points) of this initial set (called population, quite analogously to biological systems) are converted to the binary system. In the sequel, they are considered as chromosomes (actually sequences of 0 and 1).

The next step is to form pairs of these points who will be considered as parents for a "reproduction" (see the following figure)

$$\left.\begin{array}{l} 01100\,|\,100...11 \\ 00011\,|\,101...10 \end{array}\right\} \rightarrow \begin{array}{l} 01100101...10 \\ 00011100...11 \end{array}$$

parents children

"Parents" come to "reproduction" where they interchange parts of their "genetic material". (This is achieved by the so-called crossover, see the previous figure) whereas always a very small probability for a Mutation exists. (Mutation is the phenomenon where quite randomly - with a very small probability though - a 0 becomes 1 or a 1 becomes 0). Assume that every pair of "parents" gives k children.

By the reproduction the population of the "parents" are enhanced by the "children" and we have an increasement of the original population because new members were added (parents always belong to the considered population). The new population has now 2*n*+*kn* members. Then the process of natural selection is applied. According the concept of natural selection, from the *2n*+*kn* members, only 2*n* survive. These *2n* members are selected as the members with the higher values of *Q*Q, if we attempt to achieve maximization of *Q* (or with the lower values of *Q*, if we attempt to achieve minimization of *Q*Q). By repeated iterations of reproduction (under crossover and mutation) and natural selection we can find the minimum (or maximum) of *Q* as the point to which the best values of our population converge. The termination criterion is fulfilled if the mean value of *Q* in the 2n-members population is no longer improved (maximized or minimized). More detailed overviews of GAs can be found in [1], [2], [3] and [4].

So, when we have to solve a system of n equations in n unknown variables.

$$f_1(x_1, x_2, ..., x_n) = 0$$
$$f_2(x_1, x_2, ..., x_n) = 0$$
$$\vdots$$
$$f_n(x_1, x_2, ..., x_n) = 0$$

The square function $Q(x) = f_1^2 + f_2^2 + \cdots + f_n^2$ or the absolute value function $Q(x) = |f_1| + |f_2| + \cdots + |f_n|$ are defined (or in general any suitable norm of $\vec{f} = (f_1, f_2, ..., f_n)$) and our problem is min $Q(x)$

If the global minimum of $Q(x)$ is 0 at the point $(x_1^*, x_2^*, \cdots, x_n^*)$ then $x_1^*, x_2^*, \cdots, x_n^*$ is a solution of the aforementioned systems of non-linear equations. The method has also been used in [10].

4 Formulation and Solution of the General Boundary Value Problem via Collocation and Genetic Algorithm

Consider the general (second – order) initial value problem.

$$\frac{d^2y}{dx^2} = f(x, y, y') \tag{16}$$

$$\frac{dy}{dx}\Big|_{x=x_0} = y_a \tag{16.a}$$

$$\frac{dy}{dx}\Big|_{x=x_{n+1}} = y_b \tag{16.b}$$

Suppose that $\{\phi_j\}_{j=1}^{n}$ is a subset of the basis of functions (like § 2.1 or § 2.2 or § 2.3 or § 2.4), on which our solution is expanded as follows:

$$y(x) = \sum_{j=0}^{n+1} c_j \phi_j \ .$$

We can use the method of collocation again in the interval $[x_0, x_{n+1}]$.

The following system of non–linear equations is obtained

$$y(x_0) = \sum_{j=0}^{n+1} c_j \phi_j(x_0) \tag{17.0}$$

$$\sum_{j=0}^{n} c_j \phi_j''\Big|_{x_1} = f\left(x_1, \sum_{j=0}^{n+1} c_j \phi_j(x_1), \sum_{j=0}^{n+1} c_j \phi_j'(x_1)\right) \tag{17.1}$$

$$\vdots$$

$$\sum_{j=0}^{n+1} c_j \phi_j''\Big|_{x_n} = f\left(x_n, \sum_{j=0}^{n+1} c_j \phi_j(x_n), \sum_{j=0}^{n+1} c_j \phi_j'(x_n)\right) \tag{17.n}$$

$$y(x_{n+1}) = \sum_{j=0}^{n+1} c_j \phi_j(x_{n+1}) \tag{17.n+1}$$

We can solve the systems of the Equation (17.0) (17.n+1) with respect to two variables, say c_{l_1}, c_{l_2}, where $l_1, l_2 \in \{0,1,...,n\}$. Substituting these expressions of c_{l_1}, c_{l_2} into (17.1)…(15.n), we obtain a system of n non–linear equations in n unknowns: $c_0, c_1, ..., c_{l_1-1}, c_{l_1+1}, ..., c_{l_2-1}, c_{l_2+1}, ..., c_{n+1}$. These equations new equations (after the elimination of c_{l_1}, c_{l_2}) are numbered as (17a.1)…(17a.n). Please, note that these equations do not contain c_{l_1}, c_{l_2}

$$\sum_{j=0}^{n} c_j \phi_j''\Big|_{x_1} = f\left(x_1, \sum_{j=0}^{n+1} c_j \phi_j(x_1), \sum_{j=0}^{n+1} c_j \phi_j'(x_1)\right) \tag{17a.1}$$

$$\vdots$$

$$\sum_{j=0}^{n+1} c_j \phi_j''\Big|_{x_n} = f\left(x_n, \sum_{j=0}^{n+1} c_j \phi_j(x_n), \sum_{j=0}^{n+1} c_j \phi_j'(x_n)\right) \tag{17a.n}$$

The solution of (17a.1), … , (17a.n) can be obtained via a Genetic Algorithm.. The Genetic Algorithm is used to eliminate an error in (17a.1), …, (17a.n). The reason that we solve the (17.0) and (17.n+1) with respect to c_{l_1}, c_{l_2}, as well as the reason for executing the Genetic Algorithm in (17a.1),…,(17a.n) instead of (17.0),(17.1),…,(17.n),(17.n+1) is that we must satisfy the boundary value conditions (16.a) and (16.b) with zero error.

5 Concluding Remarks and Future Research

As GAs is a powerful tool for the solution of systems of non–linear equations, they can find applications in the solution of non–linear Ordinary Differential Equations. The collocation method is used and a system of non–linear equations is obtained. This system is solved by GAs. Numerical examples can outline the validity and efficiency of our proposed method.

References:

[1] Goldberg D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, Second Edition, 1989

[2] Grefenstette J.J., Optimization of control parameters for Genetic Algorithms, IEEE Trans. Systems, Man and Cybernetics, SMC 16, Jan/Feb 1986, pp. 128

[3] Eberhart R., Simpson P. and Dobbins R. (1996), *Computational Intelligence PC Tools,* AP Professionals.

[4] Kosters W.A., Kok J.N. and Floreen P., Fourier Analysis of Genetic Algorithms, Theoretical Computer Science, Elsevier, 229, 199, pp. 143-175.

[5] E. Balagusuramy, *Numerical Methods*, Tata McGraw Hill, New Delhi, 1999

[6] Nikos E. Mastorakis, “Solving Non-linear Equations via Genetic Algorithms”, Proceedings of the 6th WSEAS International Conference on Evolutionary Computing, Lisbon, Portugal, June 16-18, 2005.

[7] Ioannis F. Gonos, Lefteris I. Virirakis, Nikos E. Mastorakis, M.N.S. Swamy, "Evolutionary Design of 2-Dimensional Recursive Filters via the Computer Language GENETICA" to appear in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. (2005)

[8] Gonos I.F., Mastorakis N.E., Swamy M.N.S.: “A Genetic Algorithm Approach to the Problem of Factorization of General Multidimensional Polynomials”, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Part I, Vol. 50, No. 1, pp. 16-22, January 2003.

[9] Mastorakis N.E., Gonos I.F., Swamy M.N.S.: “Design of 2-Dimensional Recursive Filters using Genetic Algorithms”, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Part I, Vol. 50, No. 5, pp. 634-639, May 2003.

[10] Mastorakis N.E., Gonos I.F., Swamy M.N.S.: “Stability of Multidimensional Systems using Genetic Algorithms”, IEEE Transactions on Circuits and Systems, Part I, Vol. 50, No. 7, pp. 962-965, July 2003.

[11] Neil Gershenfeld (1999), *The Nature of Mathematical Modeling,* Cambridge University Press.