Genetic Algorithms and Nelder-Mead Method for the Solution of Boundary Value problems with the Collocation Method

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 computational scheme of Evolutionary Computing (Genetic Algorithms) accompanied by the Nelder-Mead method is proposed for solving Boundary Value Problems of ODEs (Ordinary Differential Equations) via the Collocation Method method.

Key-Words: - Ordinary Differential Equations, Boundary Value Problems, Finite Elements, Collocation Method, Genetic Algorithms, Evolutionary Computing, Nelder-Mead.

1 Introduction

Finite Elements' Methods are popular to engineers and scientists that are interest in the soultion of differential equations. The so called "Collocation Method" is a well known method that can be applied in Boundary Value Problems (Ordinary Differential Equations) leading the problem to the solution of a linear system of equations (when the ODE is linear) or to a system of non-linear equations (when the ODE is linear). Especially in these non-linear cases, the system of of non-linear equations is difficult to be solved and for this reason an equivalent non-linear optimization problem can be formulated. This non-linear optimization problem can be solved by several numerical schemes.

Many numerical optimization schemes have several disadvantages like inadequate accuracy, convergence to local (and not to local) minimum, low speed of convergence etc…

In [5], we have proposed a hybrid method that includes

a) Genetic Algorithm for finding rather the neiborhood of the global minimum than the global minimu itself and

b) Nelder-Mead algorithm to find the exact point of the global minimum itself.

So, with this Hybrid method of Genetic Algorithm + Nelder-Mead we combine the advantages of both methods, that are a) the convergence to the global minimum (genetic algorithm) plus b) the high accuracy of the Nelder-Mead method.

If we use only a Genetic Algorithm then we have the problem of low accuracy. If we use only Nelder-Mead, then we have the problem of the possible convergence to a local (not to the global) minimum. These disadvantages are removed in the case of our Hybrid method that combines Genetic Algorithm with Nelder-Mead method. See [5].

We recall the following definitions from the Genetic Algorithms literature:
Fitness function is the objective function we want to minimize. *Population size* specifies how many individuals there are in each generation. We can use various Fitness Scaling Options (rank, proportional, top, shift linear, etc…[16]), as well as various Selection Options (like Stochastic uniform, Remainder, Uniform, Roulette, Tournament)[16].

Fitness Scaling Options: We can use scaling functions. A Scaling function specifies the function that performs the scaling. A scaling function converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function. We have the following options: *Rank Scaling Option:* scales the raw scores based on the rank of each individual, rather than its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2 and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

Proportional Scaling Option: The Proportional Scaling makes the expectation proportional to the raw fitness score. This strategy has weaknesses when raw scores are not in a "good" range. *Top Scaling Option:* The Top Scaling scales the individuals with the highest fitness values equally. *Shift linear Scaling Option*: The shift linear scaling option scales the raw scores so that the expectation of the fittest individual is equal to a constant, which you can specify as Maximum survival rate, multiplied by the average score.

We can have also option in our Reproduction in order to determine how the genetic algorithm creates children at each new generation. For example, *Elite Counter* specifies the number of individuals that are guaranteed to survive to the next generation.
Crossover combines two individuals, or parents, to form a new individual, or child, for the next generation.

Crossover fraction specifies the fraction of the next generation, other than elite individuals, that are produced by crossover.

Scattered Crossover: Scattered Crossover creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child.

Mutation: Mutation makes small random changes in the individuals in the population, which provide genetic diversity and enable the GA to search a broader space.
Gaussian Mutation: We call that the Mutation is Gaussian if the Mutation adds a random number to each vector entry of an individual. This random number is taken from a Gaussian distribution centered on zero. The variance of this distribution can be controlled with two parameters. The Scale parameter determines the variance at the first generation. The Shrink parameter controls how variance shrinks as generations go by. If the Shrink parameter is 0, the variance is constant. If the Shrink parameter is 1, the variance shrinks to 0 linearly as the last generation is reached.

Migration is the movement of individuals between subpopulations (the best individuals from one subpopulation replace the worst individuals in another subpopulation). We can control how migration occurs by the following three parameters.

Direction of Migration: Migration can take place in one direction or two. In the so-called “Forward migration” the nth subpopulation migrates into the (n+1)'th subpopulation. while in the so-called “Both directions Migration”, the nth subpopulation migrates into both the (n-1)th and the (n+1)th subpopulation.
Migration wraps at the ends of the subpopulations. That is, the last subpopulation migrates into the first, and the first may migrate into the last. To prevent wrapping, specify a subpopulation of size zero.

Fraction of Migration is the number of the individuals that we move between the subpopulations. So, Fraction of Migration is the fraction of the smaller of the two subpopulations that moves. If individuals migrate from a subpopulation of 50 individuals into a population of 100 individuals and Fraction is 0.1, 5 individuals (0.1 * 50) migrate. Individuals that migrate from one subpopulation to another are copied. They are not removed from the source subpopulation. *Interval of Migration* counts how many generations pass between migrations.

The Nelder-Mead simplex algorithm appeared in 1965 and is now one of the most widely used methods for nonlinear unconstrained optimization [13]÷[16]. The Nelder-Mead method attempts to minimize a scalar-valued nonlinear function of n real variables using only function values, without any derivative information (explicit or implicit). The Nelder-Mead method thus falls in the general class of direct search methods. The method is described as follows:

Let $f(x)$ be the function for minimization. x is a vector in n real variables. We select $n+1$ initial points for x and we follow the steps:

Step 1. *Order*. Order the $n+1$ vertices to satisfy $f(x_1) \le f(x_2) \le \ldots \le f(x_{n+1})$, using the tie-breaking rules given below.

Step 2. *Reflect*. Compute the *reflection point* x_r from

$$x_r = \bar{x} + \rho(\bar{x} - x_{n+1}) = (1+\rho)\bar{x} - \rho x_{n+1} ,$$

where $\bar{x} = \sum_{i=1}^{n} x_i / n$ is the centroid of the n best points (all vertices except for x_{n+1}). Evaluate $f_r = f(x_r)$.

If $f_1 \le f_r < f_n$, accept the reflected point x_r and terminate the iteration.

Step 3. *Expand*. If $f_r < f_1$, calculate the *expansion point* x_e,

$$x_e = \bar{x} + \chi(x_r - \bar{x}) = \bar{x} + \rho\chi(\bar{x} - x_{n+1}) = (1+\rho\chi)\bar{x} - \rho\chi x_{n+1}$$

and evaluate $f_e = f(x_e)$. If $f_e < f_r$, accept x_e and terminate the iteration; otherwise (if $f_e \geq f_r$), accept x_r and terminate the iteration.

Step 4. *Contract*. If $f_r \geq f_n$, perform a *contraction* between $\bar{x}$ and the better of x_{n+1} and x_r.

a. *Outside*. If $f_n \leq f_r < f_{n+1}$ (i.e. x_r is strictly better than x_{n+1}), perform an *outside contraction*: calculate

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) = \bar{x} + \gamma\rho(\bar{x} - x_{n+1}) = (1+\rho\gamma)\bar{x} - \rho\gamma x_{n+1}$$

and evaluate $f_c = f(x_c)$. If $f_c \leq f_r$, accept x_c and terminate the iteration; otherwise, go to step 5 (perform a shrink).

b. *Inside*. If $f_r \geq f_{n+1}$, perform an *inside contraction*: calculate

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) = (1-\gamma)\bar{x} + \gamma x_{n+1},$$

and evaluate $f_{cc} = f(x_{cc})$. If $f_{cc} < f_{n+1}$, accept x_{cc} and terminate the iteration; otherwise, go to step 5 (perform a shrink).

Step 5. *Perform a shrink step*. Evaluate f at the n points $v_i = x_1 + \sigma(x_i - x_1)$, $i = 2, \ldots, n+1$. The (unordered) vertices of the simplex at the next iteration consist of $x_1, v_2, \ldots, v_{n+1}$.

In this paper, we try to solve Boundary Value Problems of ODEs (Ordinary Differential Equations) via the Collocation Method method
The proposed method is outlined in Section 2 with specific examples.

2 Problem formulation and Solution

Problem I:

Suppose the following linear Boundary Value Problem ([18])

$$\frac{d^2u}{dx^2} - \frac{k}{T}U + \frac{f}{T} = 0 \qquad (1)$$

$U(0) = U(L) = 0$

with:
$T = 600$
$L = 120$
$K = 0.5$
$f = 2$

Solution using Finite Elements (Collocation Method), Genetic Algorithm and Nelder-Mead.

As a trial solution, we choose for examples splines or other popular finite elements, [17]. Here, we shall use for simplicity the trial function:

$U = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_4 x^4$.

The trial solution that satisfies the boundary conditions ($\alpha_0 = 0$, $\alpha_1 = -\alpha_2 L - \alpha_4 L^3$) becomes :

$$U_R = \alpha_2(x^2 - xL) + \alpha_4(x^4 - xL^3) \qquad (2)$$

Substituting (2) into (1) gives the residual

$$R = 2\alpha_2 + 12\alpha_4 x^2 - \left(\frac{k}{T}\right)\left[\alpha_2(x^2 - xL) + \alpha_4(x^4 - xL^3)\right] + \frac{f}{T}$$

We demand "collocation" at 7 points, x = 0, 20, 40, 60, 80, 100, 120.

$$R\big|_{x=0} = 0 \qquad (3.1)$$
$$R\big|_{x=20} = 0 \qquad (3.2)$$
$$R\big|_{x=40} = 0 \qquad (3.3)$$
$$R\big|_{x=60} = 0 \qquad (3.4)$$
$$R\big|_{x=80} = 0 \qquad (3.5)$$
$$R\big|_{x=100} = 0 \qquad (3.6)$$
$$R\big|_{x=120} = 0 \qquad (3.7)$$

Equations (3.1), …, (3.7) is a system of non-linear equations in 2 unknowns: α_2 and α_4. It can be solved approximately if one demands:

$$\min_{\alpha_2,\alpha_4}\left(R^2\big|_{x=0} + R^2\big|_{x=20} + R^2\big|_{x=40} + R^2\big|_{x=60} + R^2\big|_{x=80} + R^2\big|_{x=100} + R^2\big|_{x=120}\right) \qquad (4)$$

This minimization can be achieved by GA in conjunction with Nelder-Mead.
The results are :

$\alpha_2 = -0.7105$ (5.1)
$\alpha_4 = -4 \cdot 10^{-9}$ (5.2)

In the following table, we present the parameters of the GA (Genetic Algorithm) that we used:

Parameters of GA
Population type: Double Vector
Population size: 30
Creation function: Uniform
Fitness scaling: Rank
Selection function: roulette
Reproduction: 6 – Crossover fraction 0.8
Mutation: Gaussian – Scale 1.0, Shrink 1.0
Crossover: Scattered
Migration: Both – fraction 0.2, interval: 20
Stopping criteria: 100 generation

Problem I:
Suppose now the following non-linear Boundary Value Problem.

$$\frac{du^2}{dx^2} - \frac{k}{T}U^2 + \frac{f}{T} = 0 \qquad (6)$$

U(0) = U(L) = 0

T = 600
L = 120
K = 0.5
f = 2

Solution using Finite Elements (Collocation Method), Genetic Algorithm and Nelder-Mead

We use again the trial solution U $=\alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_4 x^4$. We could choose also other popular finite elements like Splines. See [17].
The trial solution that satistfies the boundary conditions ($\alpha_0 = 0$, $\alpha_1 = -\alpha_2 L - \alpha_4 L^3$) becomes :

$$U_R = \alpha_2(x^2 - xL) + \alpha_4(x^4 - xL^3) \qquad (7)$$

Substituting (7) into (6) we obtain also the residual

$$R = 2\alpha_2 + 12\alpha_4 x^2 - \left(\frac{k}{T}\right)^2 \left[\alpha_2(x^2 - xL) + \alpha_4(x^4 - xL^3)\right]^2 + \frac{f}{T}$$

We demand again "collocation" at 7 points, x = 0, 20, 40, 60, 80, 100, 120.

$$R\big|_{x=0} = 0 \qquad (8.1)$$
$$R\big|_{x=20} = 0 \qquad (8.2)$$
$$R\big|_{x=40} = 0 \qquad (8.3)$$
$$R\big|_{x=60} = 0 \qquad (8.4)$$
$$R\big|_{x=80} = 0 \qquad (8.5)$$
$$R\big|_{x=100} = 0 \qquad (8.6)$$
$$R\big|_{x=120} = 0 \qquad (8.7)$$

Equations (8.1), …, (8.7) is a system of non-linear equations in 2 unknowns: α_2 and α_4. It can be solved approximately if one demands:

$$\min_{\alpha_2,\alpha_4}(R^2\big|_{x=0} + R^2\big|_{x=20} + R^2\big|_{x=40} + R^2\big|_{x=60} + R^2\big|_{x=80} + R^2\big|_{x=100} + R^2\big|_{x=120}) \qquad (9)$$

This minimization can be achieved by GA in conjunction with Needler – Meede.
The results are :

$$\alpha_2 = 0.5519 \cdot 10^{-5} \qquad (10.1)$$
$$\alpha_4 = 0 \qquad (10.2)$$

In the following table, we present the parameters of the GA (Genetic Algorithm) that we have used:

Parameters of GA
Population type: Double Vector
Population size: 30
Creation function: Uniform
Fitness scaling: Rank
Selection function: roulette
Reproduction: 6 – Crossover fraction 0.8
Mutation: Gaussian – Scale 1.0, Shrink 1.0
Crossover: Scattered
Migration: Both – fraction 0.2, interval: 20
Stopping criteria: 100 generation

3 Conclusion

Boundary Value Problems of ODEs are solved using Finite Elements (Collocation Method). The problem is reduced to an appropriate minimization problem which can be solved by an hybrid method that includes Genetic Algorithms and Nelder-Mead minimization technique. The method can be extended to Boundary Value Problems of Partial Differential Equations, but this will be examined in a future work.

References:

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

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

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

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

[5] Nikos E. Mastorakis, "On the Solution of Ill-Conditioned Systems of Linear and Non-Linear Equations via Genetic Algorithms (GAs) and Nelder-Mead Simplex Searchs", Proceedings of the 6th WSEAS International Conference on Evolutionary Computing, Lisbon, Portugal, June 16-18, 2005.

[6] 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)

[7] 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.

[8] 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.

[9] 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.

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

[11] Andrew Curtis and Roel Snieder: "Reconditioning inverse problems using the genetic algorithm and revised parameterization", Geophysics, Vol. 62, No. 4.pp. 1524–1532, July-August 1997.

[12] Ralf Östermark, "Solving Irregular Econometric and Mathematical Optimization Problems with a Genetic Hybrid Algorithm", Computational Economics, Volume 13 , Issue 2, pp. 103 - 115 , April 1999.

[13] Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM Journal of Optimization, Vol. 9 Number 1, pp. 112-147, 1998.

[14] J. A. Nelder and R. Mead, "A simplex method for function minimization", Computer Journal, 7 , 308-313, 1965

[15] F. H. Walters, L. R. Parker, S. L. Morgan, and S. N. Deming, Sequential Simplex Optimization, CRC Press, Boca Raton, FL, 1991

[16] Matlab, Version 7.0.0, by Math Works, Natick, MA, 1994 http://www.mathworks.com

[17] Nikos E. Mastorakis, "Numerical Solution of Non-Linear Ordinary Differential Equations via Collocation Method (Finite Elements) and Genetic Algorithms", Proceedings of the 6th WSEAS International Conference on Evolutionary Computing, Lisbon, Portugal, June 16-18, 2005.

[18] Shaoums Outline Series: Finite Elements.