American Journal of Computer Science and Engineering
2014; 1(4): 30-38
Published online November 10, 2014 (http://www.openscienceonline.com/journal/ajcse)

Performance comparison of genetic algorithm and forward (explicit) Euler method on solving the 1st order ordinary differential equations

Sambourou Massinanke, Zhang Chaozhu

Harbin Engineering University, College of Information and Communication Engineering, Laboratory Room Number: 502, Apartment of International Students, Harbin, China

Email address

smass74@yahoo.com (S. Massinanke), sam@hrbeu.edu.cn (S. Massinanke), changchaozhu@hrbeu.edu.cn (Zhang Chaozhu)

To cite this article

Sambourou Massinanke, Zhang Chaozhu. Performance Comparison of Genetic Algorithm and Forward (Explicit) Euler Method on Solving the 1st Order Ordinary Differential Equations. *American Journal of Computer Science and Engineering Research*. Vol. 1, No. 4, 2014, pp. 30-38.

Abstract

Many approximation methods have been proposed to solve ODE (Heun's Method; Midpoint; Taylor methods; Runge-Kutta; ….), some are relatively efficient, in this work we use Genetic Algorithm one famous element of Evolutionary Algorithms to solve the forward (or explicit) Euler Method (that we call simply EM) , This study explores the performance comparison of GA and EM to determine the solutions of ODEs, which implicates a search for optimal values for the unknown function in the equations that best match an Initial Value Problem (IVP).

Keyword

Forward (or Explicit) Euler Method (EM), Genetic Algorithm (GA), Continuity, Ordinary Differential Equation (ODE), Initial Value Problem (IVP)

1. Introduction

Our study presents the current scientific comprehension of the natural selection process with the aim of gaining an insight into the construction, application, and terminology of GAs (genetic algorithms). Evolution-Natural selection – is discussed in several treatises and texts. Charles Darwin's theory of evolution was founded on four primary hypotheses [1]. First, a progeny has many of the features of its parents. This hypothesis implicates that the population is stable. Secondly, variations exist in features between individuals which can be passed from one generation to the next. The third hypothesis is that only a few percentages of the progeny survive to adulthood. Fourth, which of the offspring survive depends on their inherited features. Those hypotheses combine to give the theory of natural selection. Another group of biologically-inspired techniques are Genetic Algorithms (GAs). They get their inspiration from mixing the notion of genetic recombination with the evolution theory and survival of the fittest elements of a population [2]. Beginning with a random group of candidate parameters, the learning procedure establishes better and better estimations to the optimal parameters. The genetic algorithm is mainly a search and optimization method. However, we can pose almost any practical problem as one of optimization, including a lot of environmental modeling issues. The notion of Genetic Algorithm (GA) was presented by [3] with the purpose of making computers execute what nature does. GA is one of the best methods for solving the optimization problems which involve a large search space [4]. Lots of problems in Engineering and Natural Sciences domains are solved by a scalar *differential equation* or a vector differential equation called a system of differential equations. A differential equation (DE) is an equation associating an unknown function and one or more of its derivatives..

[The building of stable and efficient methods to solve IVP is in many respect subjects [5], [6], [7], [8], [9], [10], [11].

In history, DEs (differential equation) take origin in engineering physics and chemistry, nowadays they have place in most of scientist fields: anthropology, biology, medicine, etc…

Mostly, in physical systems rise ODE (ordinary differential equation), many of them can't be integrated (solved) exactly, this is the main reason to elaborate approximation methods [12], Edwards and Penny (2000) [13], Boyce and DiPrima (2001) [14], Coombes et al. (2000) [15], Van Loan (1997) [16], Nakamura (2002) [17], Moler (2004) [18], and Gilat (2004) [19].

All these methods discretize the ODEs to produce different maps from the same equation but the aim is the same [20].

All the numerical integration methods involve different kind of errors at the integration, the effects of the errors are sometimes critical, therefore it is obvious to target least error-methods i.e. methods which are as correct as possible to assure that the solution is inside a given subset of the phase space. Moreover the subset (of the phase space) must be small [21]. EM was published in 3 volume works: [22].

During 1768-1770 Euler Method has been enhanced by many authors [23]. Despite its lacks, EM remains the basis for many higher accuracy methods [24], [25].

Unluckily, many ODEs can't be solved exactly. Therefore the ability to numerically estimate these methods is so significant. EM is a numerical technique to solve ordinary differential equations of the form:

$$\begin{cases} \beta' = f(\alpha,\beta) & p \le \alpha \le q \\ \beta(\alpha_0) = \beta_0 \end{cases}$$

An equation composed of a differential equation with an initial condition $\begin{cases} \beta' = f(\alpha,\beta) \\ \beta(\alpha_0) = \beta_0 \end{cases}$ is called a Cauchy problem

This study explores the performance comparison of GA and EM to determine the solutions of ODEs, which implicates a search for optimal values for the unknown function in the equations that best match an Initial Value Problem (IVP).

2. Euler Method (EM)

2.1. Basis

Let us consider the ordinary differential equation (ODE):

$$\begin{cases} \beta' = f(\alpha,\beta) & p \le \alpha \le q \\ \beta(\alpha_0) = \beta_0 \end{cases} \tag{1}$$

Discretization:

The central idea behind numerical methods is that of discretization. That is we partition the continuous interval $[p,q]$ by a discrete set of $N+1$ points:

$$p = \alpha_0 < \alpha_1 < \ldots < \alpha_N = q$$

The parameters

$$r_n = \alpha_{n+1} - \alpha_n,\ n = 0,1,\ldots,N-1 \tag{2}$$

are called the step-sizes. We will be often interested in using an equally spaced partition where

$$r_n = r = \frac{q-p}{N},\ n = 0,1,\ldots,N-1.$$

We will let β_n denote the numerical approximation to the exact solution $\beta(\alpha_n)$. A numerical solution of (1) consists of a set of discrete approximations $(\beta_n)_{n=0,\ldots,N}$. A numerical method is a difference equation involving a number of consecutive approximations $\beta_j, j = 0,\ldots,k$

From which we sequentially compute the sequence β_{k+n}, n = 1,…, N.

The derivation of a number of numerical methods begins by integrating (1) between α_n and α_{n+1}. This gives:

$$\int_{\alpha_n}^{\alpha_{n+1}} \frac{d\beta}{d\alpha} d\alpha = \int_{\alpha_n}^{\alpha_{n+1}} f(\alpha,\beta) d\alpha \Rightarrow$$

$$\beta(\alpha_{n+1}) - \beta(\alpha_n) = \int_{\alpha_n}^{\alpha_{n+1}} f(\alpha,\beta) d\alpha.$$

Now if we make the approximation

$$f(\alpha,\beta) \approx f(\alpha_n, \beta(\alpha_n)),\ \alpha \in (\alpha_n, \alpha_{n+1})$$

then

$$\beta(\alpha_{n+1}) - \beta(\alpha_n) \approx \int_{\alpha_n}^{\alpha_{n+1}} f(\alpha_n,\beta(\alpha_n)) d\alpha = (\alpha_{n+1} - \alpha_n) f(\alpha_n, \beta(\alpha_n))$$

therefore $\beta(\alpha_{n+1}) \approx \beta(\alpha_n) + (\alpha_{n+1} - \alpha_n) f(\alpha_n, \beta(\alpha_n))$

This suggest the numerical method:

$$\beta(\alpha_{n+1}) = \beta(\alpha_n) + (\alpha_{n+1} - \alpha_n) f(\alpha_n,\beta(\alpha_n)) = \beta(\alpha_n) + rf(\alpha_n,\beta(\alpha_n)),$$

$n = 0,\ldots,N-1$, i.e.

$$\beta_{n+1} = \beta_n + rf(\alpha_n, \beta_n),\ n = 0,\ldots,N-1 \tag{3}$$

Which is called the forward or explicit Euler Method. Note that from the initial condition

$$\beta(\alpha_0) = \beta_0$$

we can explicitly calculate β_1 by applying (3). This in turn allows us to calculate $\beta_2, \beta_3, \ldots$

The Euler Method will be closer to the exact solution as the step-size r is taken smaller ($N \to +\infty$)

2.2. Algorithm

Input: f, p, q, β_0, N .

Output: the approximate solution to $\beta' = f(\alpha, \beta(\alpha))$

With initial guess α_0 over interval $[p,q]$

- Step One: Initialization

Set $r = \frac{q-p}{N}$

Set $\beta_0 = \beta_0$

Set $\alpha_0 = p$

- Step Two: For $i=1$ to N do Step Three

  Step Three: Set $\beta_i = \beta_{i-1} + f\left(\alpha_{i-1}, \beta_{i-1}\right) \times r$

  Set $\alpha_i = \alpha_{i-1} + r$
- Step Four: Return β

Notice, algorithm returns an array of values, the i^{th} element of return array is an approximations of $\beta(\alpha)$ at $\alpha = p + ir$.

Example

Let $\beta : [1,2] \to \mathbb{R}$, $\alpha \mapsto \beta(\alpha)$ such that $\begin{cases} \beta'(\alpha) = -2\alpha^2 + 2\alpha^3 = f(\alpha,\beta) \\ \beta(1) = 0 \end{cases}$ (4)

the aim is to determine the approximate solution of (4) with $r = 0.1$.

$p=1, q=2$ $r=0.1$, $r = \frac{q-p}{N} = \frac{2-1}{N}$, then $N=10$; $\alpha_0 = 1$

$\alpha_i = \alpha_0 + ir = 1 + (0.1)i$

$$f(\alpha,\beta) = -2\alpha^2 + 2\alpha^3 + 0\beta$$

Table 1. *Result of Euler Method on Solving the ODE* $\beta : [1,2] \to \mathbb{R}$, $\alpha \mapsto \beta(\alpha)$, $\begin{cases} \beta'(\alpha) = -2\alpha^2 + 2\alpha^3 \\ \beta(1) = 0 \end{cases}$

n	α_n	β_n	$f(\alpha_n,\beta_n)$	$\beta_{n+1} = \beta_n + rf(\alpha_n,\beta_n)$
0	1	0	0	0
1	1.1	0	0.2420	0.0242
2	1.2	0.0242	0.5760	0.0818
3	1.3	0.0818	1.0140	0.1832
4	1.4	0.1832	1.5680	0.34
5	1.5	0.34	2.2500	0.565
6	1.6	0.565	3.0720	0.8722
7	1.7	0.8722	4.0460	1.2768
8	1.8	1.2768	5.1840	1.7952
9	1.9	1.7952	6.4980	2.445
10	2	2.445		

The solution given by the Euler Method (EM) is:

β_{EM} =($\beta_1 = 0$; $\beta_2 = 0.0242$; $\beta_3 = 0.0818$; $\beta_4 = 0.1832$; $\beta_5 = 0.34$; $\beta_6 = 0.565$; $\beta_7 = 0.8722$; $\beta_8 = 1.2768$; $\beta_9 = 1.7952$; $\beta_{10} = 2.445$)

3. Genetic Algorithm

3.1. Basis of GA

The aim is to determine the values of an unknown function: $\beta : [p,q] \to \mathbb{R}$, $\alpha \mapsto \beta(\alpha)$ according to a finite set of values of

$\alpha_0 = p < \alpha_1 < \ldots < \alpha_N = q$, $\alpha_i = \alpha_0 + ir$, $(i = 1,\ldots,\mathrm{N})$, (5)

$r = \frac{q-p}{N}$ we denote $\beta_i = \beta(\alpha_i)$, $(i=1,\ldots,\mathrm{N})$ the values of unknown function β, $\beta = (\beta_1,\ldots,\beta_N)$ is called the *eventual solution*.

The *population* is the set of all the eventual solutions. $\mathrm{P}(t)$ represents the population at t^{th} generation, each chromosome of the population $\beta = (\beta_1,\ldots,\beta_N)$ is characterized by its component β_i called *genetic heritage*.

In each iteration, chromosomes which are best adapted are *selected*, the surplus are discarded.

The following approximation formulas will be utilized:

- -The derivative of β function at α_i is: $\beta'(\alpha_i) \approx \frac{\beta_i - \beta_{i-1}}{r}$ (for small r).
- -*Cauchy problem discrete form*: $\frac{\beta_i - \beta_{i-1}}{r} = f(\alpha_i, \beta_i)$, $i = 1,\ldots,\mathrm{N}$, find $(\beta_1,\ldots,\beta_N)$ is our purpose.
- -Remind that general Cauchy problem is: $\begin{cases} \beta' = f(\alpha,\beta) \\ \beta(\alpha_0) = \beta_0 \end{cases}$

  where $f : D \subset \mathbb{R}^2 \to \mathbb{R}$, D open set containing (α_0, β_0).

For eventual solution $\beta = (\beta_1,\ldots,\beta_N)$, *Cauchy problem discrete form* is not verified, then we will consider the Error Formula $\left|\frac{\beta_i - \beta_{i-1}}{r} - f(\alpha_i,\beta_i)\right|$, for any eventual solution $\beta = (\beta_1,\ldots,\beta_N)$ the cost function will be:

$$H(\beta) = \sum_{i=1}^{N}\left|\frac{\beta_i - \beta_{i-1}}{r} - f(\alpha_i,\beta_i)\right|$$

The more $H(\beta)$ small the more β adapted.

For any $i = 1,\ldots,\mathrm{N}$ $\beta_i \pm \mu$ is the mutation of β_i

3.2. Genetic Algorithm Applied to Our Problem

(1) Engender a population $\mathrm{P}(i)$ associated to i^{th} generation, the initial population is a set of N chromosomes (eventual solutions), any generic chromosome is as: $\beta = (\beta_1,\ldots,\beta_N)$. Some uniform perturbation of the initial population exits i.e. $\beta_0^{(0)} = \left(\beta_0 \pm \mu_1^j,\ldots,\beta_0 \pm \mu_N^j\right)$, $j = 1,\ldots,N$, $\mu_k^j, k = 1,\ldots,\mathrm{N}$.

(2) $i \leftarrow i+1$; engender children, only the best adapted elements will survive, let $\beta = (\beta_1,\ldots,\beta_N)$ and $\gamma = (\gamma_1,\ldots,\gamma_N)$ be two individuals, $k \in \{1,\ldots,\mathrm{N}\}$ such as: $|\beta_k - \gamma_k| = \min_{l=1,\ldots,\mathrm{N}} |\beta_l - \gamma_l|$.

The algorithm will engender two children $(\beta_1,\ldots,\beta_k,\gamma_{k+1},\ldots,\gamma_N)$ and $(\gamma_1,\ldots,\gamma_k,\beta_{k+1},\ldots,\beta_N)$.

(3) the algorithm converge when $H(\beta_1,\ldots,\beta_N) = \sum_{l=1}^{N}\left|\frac{\beta_l - \beta_{l-1}}{r} - f(\alpha_l,\beta_l)\right|$ remains constant for different values of $\beta = (\beta_1,\ldots,\beta_N)$

(4) Determine the cost $H(s)$ of H at each individual $s = (s_1,\ldots,s_N)$

(5) In order to maintain the population constant, discard the surplus elements
(6) Return to (2), while stopping criteria is not attained, the stopping criteria is:
$\left|H(s_{i+1})-H(s_i)\right|<\varepsilon$ at generation i, s_i is the best individual in $\mathrm{P}\ (i)$.

3.3. Convergence of GA

An exact derivative formula (in mathematical analysis) is:
$\beta'(\alpha_i)=\lim\limits_{\alpha_i\to\alpha_{i-1}}\dfrac{\beta(\alpha_i)-\beta(\alpha_{i-1})}{\alpha_i-\alpha_{i-1}}$; ($\alpha_i=\alpha_0+ir \Rightarrow \alpha_i-\alpha_{i-1}=r$; $\alpha_i\to\alpha_{i-1} \Leftrightarrow r\to 0$), then $\beta'(\alpha_i)=\lim\limits_{r\to0}\dfrac{\beta_i-\beta_{i-1}}{r} \Leftrightarrow \beta'(\alpha_i)=\dfrac{\beta_i-\beta_{i-1}}{r}+O(r)$ with $\lim\limits_{r\to0}O(r)=0$ therefore $\exists C>0$:

$$\left|\beta'(\alpha_i)-\frac{\beta_i-\beta_{i-1}}{r}\right|\le C.r \tag{6}$$

The cost function associated H to $\beta=(\beta_1,\ldots,\beta_N)$ is:

$$H(\beta)=\sum_{l=1}^{N}\left|\frac{\beta_l-\beta_{l-1}}{r}-f(\alpha_l,\beta_l)\right|$$

$\beta=(\beta_1,\ldots,\beta_N)$ is better fitted $\Rightarrow H(\beta)$ very small.

If an eventual solution $\beta=(\beta_1,\ldots,\beta_N)$ is the limit of convergence sequence, by applying GA we get: for $\eta>0$, $\exists\beta=(\beta_1,\ldots,\beta_N)$ $H(\beta)=\sum\limits_{l=1}^{N}\left|\dfrac{\beta_l-\beta_{l-1}}{r}-f(\alpha_l,\beta_l)\right|<\eta \Rightarrow$ $\exists\beta=(\beta_1,\ldots,\beta_N)$:

$$\left|\frac{\beta_l-\beta_{l-1}}{r}-f(\alpha_l,\beta_l)\right|<r \tag{7}$$

We have: $\left|\beta'(\alpha_i)-f(\alpha_i,\beta_i)\right|\le \left|\beta'(\alpha_i)-\dfrac{\beta_i-\beta_{i-1}}{r}\right|+\left|\dfrac{\beta_l-\beta_{l-1}}{r}-f(\alpha_l,\beta_l)\right|$

Using (6) and (7) we will get $\left|\beta'(\alpha_i)-f(\alpha_i,\beta_i)\right|\le C'.r$ with $C'=C+1$,so the convergence is proven.

3.4. Convergence of EM

The necessary condition and sufficient condition for a numerical method to be convergent are stability and consistency. Stability deals with growth or decay of error as numerical computation progresses. Now we state the following theorem to discuss the convergence of EM

Theorem1

If $f(\alpha,\beta)$ satisfies a Lipchitz condition in β and is continuous in α for $0\le\alpha\le p(=\alpha_0)$ and defined a sequence β_n, $n=1,\ldots,N$ and if $\beta_0\to\beta(0)$, then $\beta_n\to\beta(\alpha)$, as $n\to+\infty$ uniformly in α where $\beta(\alpha)$ is the solution of the IVP:

$$\begin{cases}\beta'=f(\alpha,\beta)\\ \beta(\alpha_0)=\beta_0\end{cases}$$

Proof

(See the accompanied file "3-Euler Method CONVERGENCE")

Example

Let $\begin{aligned}\beta:[1,2]&\to\mathbb{R}\\ \alpha&\mapsto\beta(\alpha)\end{aligned}$ such that $\begin{cases}\beta'(\alpha)=-2\alpha^2+2\alpha^3\\ \beta(1)=0\end{cases}$ (8)

The aim is to determine the approximate solution of (8) with $r=0.1$.

Solution

We can begin by an initial population of 40 individuals and we will rejected 30 worst adapted by using the cost function

$$H(\beta_1,\ldots,\beta_{10})=\sum_{l=1}^{10}\left|\frac{\beta_l-\beta_{l-1}}{0.1}-f(\alpha_l,\beta_l)\right|$$

where $f(\alpha,\beta)=-2\alpha^2+2\alpha^3+0\beta=-2\alpha^2+2\alpha^3$

$p=1, q=2$ $r=0.1$, $r=\dfrac{q-p}{N}=\dfrac{2-1}{N}$, then $N=10$; $(\beta_1,\ldots,\beta_N)$ ie $(\beta_1,\ldots,\beta_{10})$ is the eventual solution which must be determined, $\alpha_0=1$ $\alpha_i=\alpha_0+ir=1+(0.1)i$, $i=1,\ldots,10$ are the partition of $[1,2]$, then the solution given by the Genetic Algorithm is:

$$\begin{aligned}\beta_{GA}=(&\beta_0=0,\ \beta_1=0.010,\beta_2=0.050,\beta_3=0.130,\beta_4=0.260,\\ &\beta_5=0.450,\beta_6=0.710,\beta_7=1.070,\beta_8=1.530,\\ &\beta_9=2.110,\beta_{10}=2.830)\end{aligned}$$

4. Theorem 1

If $f(\alpha,\beta)$ and $\dfrac{\partial f}{\partial\beta}(\alpha,\beta)$ are both continuous functions of α and β in a region $0<|\alpha-\alpha_0|<p$ and $0<|\beta-\beta_0|<q$ then there exists a unique solution $\beta=\beta(\alpha)$ in the interval $0<|\alpha-\alpha_0|<r\le p$, that satisfies the initial value problem (IVP): $\begin{cases}\dfrac{d\beta}{d\alpha}=f(\alpha,\beta)\\ \beta(\alpha_0)=\beta_0\end{cases}$

(file: 1-MATH10232_see theorem2.1, there is no information to reference this file, that is why I send u it)

Remark: our example function satisfies the theorem.

$$\begin{aligned}\beta:[1,2]&\to\mathbb{R}\\ \alpha&\mapsto\beta(\alpha)\end{aligned}\quad\begin{cases}\beta'(\alpha)=f(\alpha,\beta)=-2\alpha^2+2\alpha^3+0\beta=-2\alpha^2+2\alpha^3\\ \beta(1)=0\end{cases}$$

$$f(\alpha,\beta)=-2\alpha^2+2\alpha^3+0\beta=-2\alpha^2+2\alpha^3;$$

$\frac{\partial f}{\partial \beta}(\alpha,\beta) = -2\alpha^2 + 2\alpha^3$ are both continuous (because polynomial functions) in $0 < |\alpha - 1| < 1$ and $0 < |\beta - 0| < 2$

5. Theorem2 (Error in EM)

Suppose that $f(\alpha,\beta), \frac{\partial f(\alpha,\beta)}{\partial \alpha}$ and $\frac{\partial f(\alpha,\beta)}{\partial \beta}$ are continuous and bounded functions on the rectangle $A = [\mathrm{g,m}] \times \mathbb{R}$ and that that the interval $[\mathrm{p,q}]$ satisfies $g < p < q < m$. Let $Error_k = \beta(\alpha_k) - \beta_k$ (β_k denote the numerical approximation to the exact solution $\beta(\alpha_k)$) denote the error at step k in applying Euler's Method with N steps of length r to the differential equation $\beta' = f(\alpha,\beta)$ on the interval $[\mathrm{p,q}]$, with initial condition $\beta(\alpha_0) = \beta_0$.

Then $|Error_k| \le r\frac{D}{2C}\left(e^{(\alpha_k - p)C} - 1\right) \le r\frac{D}{2C}\left(e^{(q-p)C} - 1\right)$ for $k = 0,1,....,\mathrm{N}$ where the constants C and D are given by:

$$C = \max_{(\alpha,\beta)\in A}\left|\frac{\partial f}{\partial \beta}(\alpha,\beta)\right|,\ D = \max_{\alpha\in[\mathrm{p,q}]}\left|\beta''(\alpha)\right|.$$

(file: 2-differential_ seeTheorem 12.12, there is no information to reference this file, that is why I send u it)

We have: $|Error_k| \le r\frac{D}{2C}\left(e^{(q-p)C} - 1\right)$ for any $k = 0,1,...,N \Rightarrow$

$$\sup_{k\in\{0,1,...,N\}}|Error_k| \le \sup_{k\in\{0,1,...,N\}} r\frac{D}{2C}\left(e^{(q-p)C} - 1\right) = r\frac{D}{2C}\left(e^{(q-p)C} - 1\right)$$

if we will admit:

$Error_{EM} = r\frac{D}{2C}\left(e^{(q-p)C} - 1\right) = r\frac{D}{2C}\left(e^{rNC} - 1\right)$; ($r = \frac{q-p}{N}$) and

$$Error_{GA} = H(\beta) = \sum_{i=1}^{N}\left|\frac{\beta_i - \beta_{i-1}}{r} - f(\alpha_i,\beta_i)\right|$$

6. Performance Comparison of GA and EM

6.1. Case of the Example

Computation of $Error_{EM}$ applied on β_{EM} noted $Error_{EM}(\beta_{EM})$

$Error_{EM} = r\frac{D}{2C}\left(e^{(q-p)C} - 1\right)$, This Error doesn't depend on α_i, β_i and f

$f(\alpha,\beta) = -2\alpha^2 + 2\alpha^3 + 0\beta = -2\alpha^2 + 2\alpha^3$; $C = \max_{(\alpha,\beta)\in A}\left|\frac{\partial f}{\partial \beta}(\alpha,\beta)\right|$;

$D = \max_{\alpha\in[\mathrm{p,q}]}\left|\beta''(\alpha)\right|$ it is easy to find C, D ; $C = 8; D = 16$;

$$Error_{EM} = r\frac{D}{2C}\left(e^{(q-p)C} - 1\right) = \frac{e^8 - 1}{10} = 282.3$$

$Error_{EM}(\beta_{EM}) = 282.3$

Computation of $Error_{GA}$ applied on β_{GA} noted $Error_{GA}(\beta_{GA})$

$$\beta_{GA} = (\beta_0 = 0,\ \beta_1 = 0.010, \beta_2 = 0.050, \beta_3 = 0.130, \beta_4 = 0.260, \beta_5 = 0.450, \beta_6 = 0.710, \beta_7 = 1.070, \beta_8 = 1.530, \beta_9 = 2.110, \beta_{10} = 2.830)$$

$$Error_{GA} = H(\beta) = \sum_{i=1}^{N}\left|\frac{\beta_i - \beta_{i-1}}{r} - f(\alpha_i,\beta_i)\right| \quad ; \quad \alpha_0 = 1$$

$\alpha_i = \alpha_0 + ir = 1 + (0.1)i$, $i = 1,...,10$; $N = 10$; if we replace the letters by their values $Error_{GA}(\beta_{GA}) = 4.767$.

$Error_{EM}(\beta_{EM}) > Error_{GA}(\beta_{GA}) \Rightarrow$ GA is more accurate than EM (in this example)

6.2. General Case

Suppose theorem 1 and theorem 2 assumptions satisfied, for the solution $\beta = (\beta_1,...,\beta_N)$ of the ordinary differential equation: $\begin{cases}\beta' = f(\alpha,\beta) & p \le \alpha \le q \\ \beta(\alpha_0) = \beta_0\end{cases}$

we compare the Errors

$$Error_{EM} = r\frac{D}{2C}\left(e^{(q-p)C} - 1\right);$$

$$Error_{GA} = H(\beta) = \sum_{i=1}^{N}\left|\frac{\beta_i - \beta_{i-1}}{r} - f(\alpha_i,\beta_i)\right|$$

Remind that: $r = \frac{q-p}{N}$; N is the number of the subdivisions of the interval $[\mathrm{p,q}]$ then $N \ge 1; r > 0$.

If we replace $q - p$ by its value in $Error_{EM}$ we get $Error_{EM} = r\frac{D}{2C}\left(e^{rNC} - 1\right)$; and

$$Error_{GA} = H(\beta) = \sum_{i=1}^{N}\left|\frac{\beta_i - \beta_{i-1}}{r} - f(\alpha_i,\beta_i)\right|$$

$D, C, \alpha_i, \beta_i, f(\alpha_i,\beta_i)(i = 1,...,N)$ are all constant, the general case must be done on r, N

We first vary the value of r in $]0,+\infty[$

In this case $Error_{EM}$ and $Error_{GA}$ become functions or r;

$Error_{EM} = h_1(r) = r\frac{D}{2C}\left(e^{rNC} - 1\right)$ and

$$Error_{GA} = h_2(r) = \sum_{i=1}^{N}\left|\frac{\beta_i - \beta_{i-1}}{r} - f(\alpha_i,\beta_i)\right|$$

$h_1(r)$ and $h_2(r)$ are both continuous function on

$r\in]0,+\infty[$; $\lim_{r\to+\infty}\frac{h_1}{h_2}=\lim_{r\to+\infty}\frac{r\frac{D}{2C}e^{rNC}}{\sum_{i=1}^{N}\left|\frac{\beta_i-\beta_{i-1}}{r}-f(\alpha_i,\beta_i)\right|}=$

$\lim_{r\to+\infty}\frac{D}{2C}\times\frac{re^{rNC}}{\sum_{i=1}^{N}\left|f(\alpha_i,\beta_i)\right|}=\lim_{r\to+\infty}\frac{D}{2C}\times\frac{re^{rNC}}{L}$ where

$L=\sum_{i=1}^{N}\left|f(\alpha_i,\beta_i)\right|=const$,therefore

$\lim_{r\to+\infty}\frac{h_1}{h_2}=\frac{D}{2LC}\times\lim_{r\to+\infty}re^{rNC}=+\infty\Rightarrow$

that means $\exists R>0$, $\forall r\in]0,+\infty[$, $r>R\Rightarrow h_1(r)>h_2(r)$ then $Error_{EM}>Error_{GA}$ then GA is more accuracy than EM

Let us vary the value of N *in* $[1,+\infty[$

$$\begin{cases} Error_{EM}=h_1(N)=r\frac{D}{2C}\left(e^{rNC}-1\right) \\ Error_{GA}=h_2(N)=\sum_{i=1}^{N}\left|\frac{\beta_i-\beta_{i-1}}{r}-f(\alpha_i,\beta_i)\right| \end{cases}$$

First both functions are continuous on $[1,+\infty[$

$$\lim_{N\to+\infty}h_1(N)=\lim_{N\to+\infty}r\frac{D}{2C}\left(e^{rNC}\right)=+\infty \;;$$

$$\lim_{N\to+\infty}h_2(N)=\lim_{N\to+\infty}\sum_{i=1}^{N}\left|\frac{\beta_i-\beta_{i-1}}{r}-f(\alpha_i,\beta_i)\right|$$
$$=\lim_{N\to+\infty}\left(\left|\frac{\beta_0-\beta_1}{r}-f(\alpha_1,\beta_1)\right|+\left|\frac{\beta_1-\beta_2}{r}-f(\alpha_2,\beta_2)\right|+\ldots+\left|\frac{\beta_{N-1}-\beta_N}{r}-f(\alpha_N,\beta_N)\right|\right)$$

$=L$, there are two possibilities: $0<L<+\infty$ *or* $L=+\infty$

1st case: $0<L<+\infty$

$\lim_{N\to+\infty}h_1(N)=+\infty$; $\lim_{N\to+\infty}h_2(N)=L>0\Rightarrow$ mathematically that means $\exists R>0,\forall r\in[1,+\infty[$, $r>R\Rightarrow h_1(N)>h_2(N)$ then $Error_{EM}>Error_{GA}$ then GA is more efficient than EM

2nd case: $L=+\infty$

$\lim_{N\to+\infty}h_1(N)=+\infty$; $\lim_{N\to+\infty}h_2(N)=L=+\infty$, we can process like this:

$$\lim_{N\to+\infty}\frac{h_1(N)}{h_2(N)}=\lim_{N\to+\infty}\frac{r\frac{D}{2C}\left(e^{rNC}\right)}{\sum_{i=1}^{N}\left|\frac{\beta_i-\beta_{i-1}}{r}-f(\alpha_i,\beta_i)\right|}=$$

$$\lim_{N\to+\infty}\frac{r\frac{D}{2C}\left(e^{rNC}\right)}{\left|\frac{\beta_0-\beta_1}{r}-f(\alpha_1,\beta_1)\right|+\left|\frac{\beta_1-\beta_2}{r}-f(\alpha_2,\beta_2)\right|+\ldots+\left|\frac{\beta_{N-1}-\beta_N}{r}-f(\alpha_N,\beta_N)\right|}=$$

$$\lim_{N\to+\infty}\frac{e^{rNC}}{\left|\frac{\beta_0-\beta_1}{r}-f(\alpha_1,\beta_1)\right|+\left|\frac{\beta_1-\beta_2}{r}-f(\alpha_2,\beta_2)\right|+\ldots+\left|\frac{\beta_{N-1}-\beta_N}{r}-f(\alpha_N,\beta_N)\right|}=$$

$$\lim_{N\to+\infty}\frac{e^{aN}}{\left|\frac{\beta_0-\beta_1}{r}-f(\alpha_1,\beta_1)\right|+\left|\frac{\beta_1-\beta_2}{r}-f(\alpha_2,\beta_2)\right|+\ldots+\left|\frac{\beta_{N-1}-\beta_N}{r}-f(\alpha_N,\beta_N)\right|}$$

(with $a=rC>0$)

In Mathematical Analysis there is a theorem which states:

Theorem 3

If $f(x)=e^x$ (Exponential function); for any Polynomial function $P(x)=\sum_{i=0}^{N}a_ix^i$ $\left(a_N\neq 0; N\neq 0;\quad N=\deg ree(P)\right)$; then:

$$\lim_{x\to+\infty}\frac{f(x)}{P(x)}=\lim_{x\to+\infty}\frac{e^x}{a_Nx^N}=\begin{cases}+\infty\; if\; a_N>0\\ -\infty\; if\; a_N>0\end{cases}$$

This theorem allows us to conclude that: $\lim_{N\to+\infty}\frac{h_1(N)}{h_2(N)}=+\infty$

$\lim_{N\to+\infty}\frac{h_1(N)}{h_2(N)}=+\infty\Rightarrow$ mathematically that means $\exists R>0$, $\forall r\in[1,+\infty[$, $r>R\Rightarrow h_1(N)>h_2(N)$ then $Error_{EM}>Error_{GA}$ then GA is more efficient than EM

We have just showed that in any case GA is more efficient than EM.

6.3. Experimental Study

Experiment 1:

The *Example (above)*

$\beta:[1,2]\to\mathbb{R}$, $\alpha\mapsto\beta(\alpha)$ such that $\begin{cases}\beta'(\alpha)=-2\alpha^2+2\alpha^3=f(\alpha,\beta)\\ \beta(1)=0\end{cases}$ (eq)

the aim is to determine the approximate solution of (eq) with.

$p=1,q=2$, $N=10$ $\quad r=\frac{q-p}{N}=\frac{1}{10}=0.1$

$\beta'=-2\alpha^2+2\alpha^3\Rightarrow\beta=\frac{-2}{3}\alpha^3+\frac{2}{4}\beta^4+C$ where C is any real number; $\beta(1)=0\Rightarrow C=\frac{1}{6}$, then

$\beta:[1,2]\to\mathbb{R}$, $\alpha\mapsto\beta=\frac{-2}{3}\alpha^3+\frac{2}{4}\alpha^4+\frac{1}{6}$; $\alpha_n=\alpha_{n-1}+r=\alpha_0+nr$ where $n=0,1,\ldots,10$ $\beta_n=\beta(\alpha_n), n=0,1,\ldots,10$; the exact solution will be

$$\beta_{Exact}=\begin{pmatrix}\beta_0=0,\beta_1=0.0114,\beta_2=0.0515,\beta_3=0.1301,\\ \beta_4=0.2581,\beta_5=0.4479\\ \beta_6=0.7128,\beta_7=1.0674,\\ \beta_7=1.0674,\beta_8=1.5275,\\ \beta_9=2.11,\beta_{10}=2.8333\end{pmatrix}$$

Table 2. *Performance comparison of GA and EM on solving the example:*

Genetic Algorithm				**Euler-Method**			**Conclusion**
n	$\beta_{GA} = (\beta_n)_{n=0,1,\ldots,10}$	β_{EXACT}	$Error(GA) = \lvert\beta_{EXACT} - \beta_{GA}\rvert$	β_{EM}	β_{EXACT}	$Error(EM) = \lvert\beta_{EXACT} - \beta_{EM}\rvert$	**The best= The least**
0	0	0	0	0	0	0	EM= GA
1	0.010	0.0114	0.00140	0	0.0114	0.0114	GA better
2	0.050	0.0515	0.0015	0.0242	0.0515	0.0273	GA better
3	0.130	0.1301	0.0001	0.0818	0.1301	0.0483	GA better
4	0.260	0.2581	0.0019	0.1832	0.1301	0.0531	GA better
5	0.45	0.4479	0.0021	0.34	0.2581	0.0819	GA better
6	0.710	0.7128	0.0028	0.565	0.7128	0.1478	GA better
7	1.070	1.0674	0.0026	0.8722	1.0674	0.1952	GA better
8	1.530	1.5275	0.0025	1.2768	1.5275	0.2507	GA better
9	2.110	2.11	0	1.7952	2.11	0.3148	GA better
10	2.830	2.8333	0.0033	2.445	2.8333	0.3883	GA better

Experiment 2:

Table 3. *Performance comparison of GA and EM on solving* $\begin{cases} \beta : [0,1] \to \mathbb{R} \\ \beta' = -4.32\beta \quad r = 0.01 \\ \beta(0) = 1 \end{cases}$

Genetic Algorithm				**Euler-Method**			**Conclusion**
n	β_{GA}	β_{EXACT}	$Error(GA) = \lvert\beta_{EXACT} - \beta_{GA}\rvert$	β_{EM}	β_{EXACT}	$Error(EM) = \lvert\beta_{EXACT} - \beta_{EM}\rvert$	**The best= The least**
0	1	1	0	1	1	0	EM= GA
1	0.647	0.64921	0.00221	0.643	0.64921	0.00621	GA better
2	0.420	0.42147	0.00127	0.41345	0.42147	0.008	GA better
3	0.267	0.27362	0.00662	0.26585	0.27362	0.00777	GA better
4	0.175	0.17764	0.00264	0.17094	0.17764	0.0067	GA better
5	0.108	0.17764	0.06964	0.10992	0.11533	0.00541	EM better
6	0.072	0.07487	0.00287	0.07067	0.07487	0.0042	GA better
7	0.047	0.04860	0.0016	0.04544	0.04860	0.00316	GA better
8	0.030	0.03155	0.00155	0.02922	0.03155	0.00223	GA better
9	0.021	0.02048	0.00052	0.01878	0.02048	0.0017	GA better
10	0.0129	0.01330	0.0004	0.01208	0.01330	0.00122	GA better

Experiment 3:

Table 4. *Performance comparison of GA and EM on solving* $\begin{cases} \beta : [0,2.5] \to \mathbb{R} \\ \beta'(\alpha) = -1.2\beta + 7e^{-0.3\alpha} \\ \beta(0) = 3 \end{cases}$

Genetic Algorithm				Euler-Method			Conclusion
n	β_{GA}	β_{EXACT}	$Error(GA) = \lvert\beta_{EXACT} - \beta_{GA}\rvert$	β_{EM}	β_{EXACT}	$Error(EM) = \lvert\beta_{EXACT} - \beta_{EM}\rvert$	The best= The least
0	3	3	0	3	3	0	EM =GA
1	4.060	4.072	0.012	4.7	4.072	0.628	GA better
2	4.330	4.323	0.007	4.893	4.323	0.57	GA better
3	4.201	4.170	0.031	4.55	4.170	0.38	GA better
4	3.801	3.835	0.03	4.052	3.835	0.217	GA better
5	3.450	3.436	0.014	3.542	3.436	0.106	GA better

Experiment 4:

Table 5. *Performance comparison of GA and EM on solving* $\begin{cases} \beta:[0,4]\to\mathbb{R} \\ \beta'(\alpha)=-2\alpha^3+12\alpha^2-20\alpha+8.5 \\ \beta(0)=1 \end{cases}$, $r=0.5$

Genetic Algorithm				**Euler-Method**			**Conclusion**
n	β_{GA}	β_{EXACT}	$Error(GA)=\lvert\beta_{EXACT}-\beta_{GA}\rvert$	β_{EM}	β_{EXACT}	$Error(EM)=\lvert\beta_{EXACT}-\beta_{EM}\rvert$	**The best= The least**
0	1	1	0	1	1	0	GA better
1	4.210	3.2188	0.9912	5.25	3.2188	2.0312	GA better
2	3.511	3	0.511	5.875	3	2.875	GA better
3	3.22	2.2188	1.0012	5.125	2.2188	2.9062	GA better
4	1.910	2	0.09	4.500	2	2.5	GA better
5	2.66	2.7188	0.0588	4.750	2.7188	2.0312	GA better
6	3.11	4	0.89	5.875	4	1.875	GA better
7	5.4	4.7187	0.6813	7.125	4.7187	2.4063	GA better
8	2.9	3	0.1	7	3	4	GA better

Experiment 5:

Table 6. *Performance comparison of GA and EM on solving* $\begin{cases} \beta:[0,1]\to\mathbb{R} \\ \beta'(\alpha)=-\beta+\alpha \\ \beta(0)=1 \end{cases}$, $r=0.2$

Genetic Algorithm				**Euler-Method**			**Conclusion**
n	β_{GA}	β_{EXACT}	$Error(GA)=\lvert\beta_{EXACT}-\beta_{GA}\rvert$	β_{EM}	β_{EXACT}	$Error(EM)=\lvert\beta_{EXACT}-\beta_{EM}\rvert$	**The best= The least**
0	1	1	0	1	1	0	EM =GA
1	0.82	0.837	0.017	0.8	0.837	0.037	GA better
2	0.689	0.741	0.052	0.68	0.741	0.061	GA better
3	0.620	0.698	0.078	0.624	0.698	0.074	EM better
4	0.65	0.699	0.049	0.619	0.699	0.08	GA better
5	0.71	0.736	0.026	0.655	0.736	0.081	GA better

Experiment 6:

Table 7. *Performance comparison of GA and EM on solving* $\begin{cases} \beta:[0,0.5]\to\mathbb{R} \\ \beta'(\alpha)=\alpha\beta \\ \beta(0)=1 \end{cases}$ $r=0.1$

Genetic Algorithm				**Euler-Method**			**Conclusion**
n	β_{GA}	β_{EXACT}	$Error(GA)=\lvert\beta_{EXACT}-\beta_{GA}\rvert$	β_{EM}	β_{EXACT}	$Error(EM)=\lvert\beta_{EXACT}-\beta_{EM}\rvert$	**The best= The least**
0	1	1	0	1	1	0	EM= GA
1	1.004	1.00501	0.00101	1	1.00501	0.00501	GA better
2	1.012	1.0202	0.0082	1.01	1.0202	0.0102	GA better
3	1.048	1.04603	0.00197	1.0302	1.04603	0.01	GA better
4	1.070	1.08329	0.01329	1.061106	1.08329	0.022184	GA better
5	1.102	1.13315	0.03115	1.1035524	1.13315	0.0295976	EM better

7. Conclusion

In this work we have studied GA and EM, we have applied both to solve the same equation, we have compared mathematically and experimentally proven that GA and EM and the remark is:

In almost all the case GA outperforms EM in solving the ODE, Genetic Algorithms is a strong tool for many problems in scientific computation. It can solve not only ODE but also can be applied to solve problems of numerical analysis such System of linear equations, polynomial factorization, Travelling of Salesman Person, Knapsack Problem etc.

So we can conclude that GA is more efficient than EM in solving of the ODE of the form:

$$\begin{cases} \beta' = f(\alpha,\beta) \quad p\le\alpha\le q \\ \beta(\alpha_0)=\beta_0 \end{cases}.$$

GA remains a powerful technique to tackle problems in which there is no known method to get the solution.

Acknowledgements

This work is sponsored by the international exchange program of Harbin Engineering University for innovation oriented talents cultivation.

References

[1] V. L. C. M. A. P. a. J. K. W. S. E. Haupt, " Environmental Science Models and Artificial Intelligence, Artificial Intelligence Methods in the Environmental Sciences.," Springer Science, pp. (3-14, 103-126), 2009.

[2] C. DARWIN:, The origin of species by means of natural selection, or the Preservation of Favoured Races in the Struggle for Life,, London: John Murray,, 1859.

[3] J.H. Holland: *Adaptation in Natural and Artificial Systems*, University of Michigan Press, USA, 1975.

[4] M. M. Ibrahiem El-Emary and M. 2Mona Abd El-Kareem: *Towards Using Genetic Algorithm for Solving Nonlinear Equation Systems* World Applied Sciences Journal 5 (3): 282-289, 2008.

[5] K.E. Brenan, S.L. Campbell, and L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, SIAM, Philadelphia, 1995. [5] K.E. Brenan, S.L. Campbell, and L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, SIAM, Philadelphia, 1995.

[6] J. Butcher, The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods, Wiley, 1987.

[7] C. W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, New Jersey, 1971.

[8] E. Hairer, S. P. Norsett and G.Wanner, Solving Ordinary Differential Equations I, Non-Stiff Problems, Springer Verlag, Berlin, 1987.

[9] E. Hairer, G.Wanner, Solving Ordinary Differential Equations II, Springer, 1996.

[10] Arieh Iserles, A First Course in the Numerical Analysis of Differential Equations, Cambridge University Press, 1996.

[11] J. D. Lambert, Numerical Methods for Ordinary Differential Equations, Wiley, 1991.

[12] Rattenbury N., Almost Runge-Kutta methods for stiff and non-staiff problems, Ph.D Dissertation, The University of Auckland, New Zealand, 2005.

[13] C. H. Edwards and D. E. Penny. Di®erential Equations and Boundary Value Problems: Computing and Modeling. Prentice Hall, 2000.

[14] W. E. Boyce and R. C. DiPrima. Elementary Di®erential Equations and Boundary Value Problems. John Wiley and Sons, 2001.

[15] K. R. Coombes, B. R. Hunt, R. L. Lipsman, J. E. Osborn, and G. J. Stuck. Di®erential Equations with MATLAB. John Wiley and Sons, 2000.

[16] C. F. Van Loan. Introduction to Scienti¯c Computing. Prentice Hall, 1997.

[17] S. Nakamura. Numerical Analysis with MATLAB. Prentice Hall, 2002.

[18] C. B. Moler. Numerical Computing with MATLAB. Siam, 2004.

[19] A. Gilat. MATLAB: An introduction with Applications. John Wiley and Sons, 2004.

[20] Julyan E. H. C., Piro O., The dynamics of Runge-Kutta methods, Int'1 J Bifur and Chaos 2, 1992.

[21] E.M. Alessi, A. Farrs, . Jorba, C. Sim, A. Vieiro 2008. Efficient Usage of Self Validated Integrators for Space Applications University of Barcelona Final Report Departament de Matematica Aplicada i Analisi Universitat de Barcelona.

[22] Euler, L. 1913:De integratione aequationum differentialium per approximationem, In Opera Omnia, 1st series, Vol. 11, *Institutiones Calculi Integralis* , Teubner, Leipzig and Berlin, 424434.

[23] Euler, H. 1768: Institutiones calculi integralis . Volumen Primum, Opera Omnia, Vol. XI, B. G. Teubneri Lipsiae et Berolini MCMXIII.

[24] Lee, J. H. J. 2004: Numerical Methods for Ordinary Differential Systems: A Survey of Some Standard Methods. M. Sc. Thesis, University of Auckland, Auckland, New Zealand.

[25] Runge, C. Uber die numerische Auflosung von differntialglechungen, Math. Ann. 46 (1895) 167-178.