International Journal of Mathematics and
Computer Science, **9**(2014), no. 1, 1–10

Fibonacci Scheme On Gradient Method For Some Control Problems

J. O. Omolehin, I. Abdullahi, K. Rauf and P. L. Dickmu

Department of Mathematics
University of Ilorin
Ilorin, Nigeria

email: omolehin_joseph@yahoo.com, iabdullahi94@gmail.com,
krauf@unilorin.edu.ng, pawahndukum@yahoo.co.uk

(Received November 10, 2013, Accepted January 13, 2014)

Abstract

We considered Computational results with Fibonacci scheme for Gradient method. In Particular, Fibonacci scheme is used as against the arbitrary perturbation term, lambda (λ) in Gradient method. It is proved that Fibonacci technique converges to the solution if the interval of uncertainty is known and N is large for the Fibonacci sequence in Gradient Method.

1 Introduction

The theory of optimization is significant and applicable to problem involving decision making in all field of endeavor. This is governed by the desire to make the best decision. Therefore, optimization theory and methods deal with selecting the best alternative in the sense of the given objective function [6].
In an optimization problem we seek values of the variables that lead to an optimal value of the function that is to be optimized. Thus, some functions of the variables that describe the problem must be maximized(or minimized) by a suitable choice of the variable within some permitted choice of the variable within some permitted feasible region [3].

Key words and phrases: Gradient Method, Fibonacci Scheme, Perturbation, Convergence.
AMS (MOS) Subject Classifications: 49J20, 47N10 34M60, 11B39, 46N10, 65K05.
ISSN 1814-0424 2014, http://ijmcs.future-in-tech.net

2 Fibonacci Scheme

The Fibonacci method can be used to find the minimum of a function of one variable even if the function is not continuous. This method makes use of the sequence of Fibonacci numbers. These numbers are defined as $F_0 = F_1 = 1$ $F_n = F_{n-1} + F_{n-2}$, n = 2 , 3 , 4 , . . . which yield the sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

3 Gradient Method

Consider the function $f : \mathbb{R}^n \to \mathbb{R}$ which is continuously differentiable in some domain $D \in \mathbb{R}^n$ and it is assumed that f assumes a local minimum value in D at a point $x \in \mathbb{D}^o$, where $\mathbb{D}^o$ is the interior of D [8, 10, 2]. In order to construct the formula used in what is commonly called the gradient method, we let λ be a positive number and consider the tailor series expansion truncated at the second term

$$f(x - \lambda\frac{\partial f}{\partial x}(x)^{'}) = f(x) + \frac{\partial f}{\partial x}(x)(-\lambda\frac{\partial f}{\partial x}(x)^{'}) + \theta(\lambda)$$

$$= f(x) - \lambda||\frac{\partial f}{\partial x}(x)||_e^2 + \theta(\lambda).$$

If $\frac{\partial f}{\partial x}(x) \neq 0$ then for sufficiently small $\lambda > 0$ we clearly have $f(x - \lambda\frac{\partial f}{\partial x}(x)^{'}) < f(x)$ Thus, if we are searching for a minimum of f the point $x - \lambda\frac{\partial f}{\partial x}(x)^{'}$ is an improvement over the point x if $\frac{\partial f}{\partial x}(x) \neq 0$ and λ is positive and in the neighborhood of zero. By repeated construction of new points in this manner we may hope to approach x^*, That is, local minimum of f in $\mathbb{D}$. The gradient consists in the construction of a sequence $\{x\}$ of points in $\mathbb{R}^n$ by the recursion equation

$$x_{i+1} = x_i - \lambda\frac{\partial f}{\partial x}(x_i^{'}), i = 0, 1, 2, \ldots$$

where x_o is the initial guess value [1]. The convergence rate of gradient method based on the choice of λ has been seriously consider in [1]. The major problem associated with gradient method algorithm is the choice of λ. In this work, we choose λ as the inverse of the Fibonacci number ($\frac{1}{F_n}, n = 2, 3, \ldots)$ to observe the convergence behavior. That is, at each iteration, the value of

λ will be varied, while In [1] the value of λ is taken to be constant at each iteration.

4 Main result

In this section, we shall solve the problem considered in [1] and then compare the convergence rate of our method with the result in [1].
Example. The Cosmopolitan Encyclopedia Co. normally sells its product for cash or for no money down with cost spread over twenty four equal payments. The basic selling price is the same in either case but a service charge equal to a certain percent of the selling price is added to time payment accounts. Thus the company's income comes from two sources, the price charged for the encyclopedias and the service charges collected on time payment.
The encyclopedias cost \$ 100 to produce. We will let the selling price be \$ 100(1+x), where x is , of course, positive. We let the service charge on time payments by y percent of the selling price.
Experience indicates the following to be true:
Total sales are proportional to $\frac{1}{1+x+x^2}$ in the price range under

$$f(x,y) = \frac{x}{1+x+x^2} + \frac{(y-\frac{y^2}{20})(x+\frac{1}{2})}{1+x+x^2}.$$

We compute the partial derivatives of f:

$$\frac{\partial f}{\partial x} = \frac{1}{1+x+x^2}(1+y-\frac{y^2}{20}) - \frac{2x+1}{(1+x+x^2)^2}(x+(y-\frac{y^2}{20})(x+\frac{1}{2})),$$

$$\frac{\partial f}{\partial y} = \frac{(x+\frac{1}{2})}{1+x+x^2}(1-\frac{y}{10}).$$

In this particular case the equations $\frac{\partial f}{\partial x} = 0, \frac{\partial f}{\partial y} = 0$ can be solved by hand to have y = 10 and x = 0.45. Thus the selling price should be \$ 145 and the service charge should be \$ 10% . For details of Arithematics see [1]. Let us see how the gradient method will bring us to the same result. Maximizing f(x,y) is the same as minimizing -f(x,y). The gradient method for minimizing -f(x,y) is

$$x_{i+1} = x_i - \lambda\frac{\partial(-f)}{\partial x}(x_i,y_i) = x_i + \lambda\frac{\partial f}{\partial x}(x_i,y_i),$$

$$y_{i+1} = y_i - \lambda \frac{\partial(-f)}{\partial y}(x_i, y_i) = x_i + \lambda \frac{\partial f}{\partial y}(x_i, y_i).$$

4.1 Analysis of the Tables

The table 1 is the work carried out in [1] by taking the value of λ as constant, that is $\lambda = \frac{1}{2}$ in each iteration to obtained the optimum solution. In table 2 when they choose $\lambda = 1$, they succeeded in speeding up the convergence of the y_i toward its optimum while the x_i values show no sign of convergence. In table 3, we have been able to show that the same optimum can be obtained using Fibonacci sequence. That is, we vary the value of λ from one iteration to the next using Fibonacci scheme $\lambda = \frac{1}{F_n}, n = 2, 3, \ldots$. It is observed that from one iteration to another, the value of x and y decreases and increases respectively. Also, in table 4 we choose initial values far away from their optimum, it still show sign of convergence. Which makes our results better than the results in [1].

5 Conclusion

The work carried out in [1]. They choose initial points x = 1, y = 5 and $\lambda = \frac{1}{2}$ and computed up to 100 iterations to get x = 0.45342 and y = 9.72745 and when they choose $\lambda = 1$ with x = 1 and y = 5 and computed up to 50 iterations to get y = 9.60221 while x show no sign of convergence at all. We now observed that, chosen initial points x = 1, y = 5 and $\lambda = \frac{1}{2}$ the convergence obtained at 408th and 948th iteration for x and y respectively. That is, x = 0.453358875 and y = 10.0000. Also, chosen $\lambda = 1$ with x = 1 and y = 5, the exact solution obtained at 577th iteration for y and x showed no sign of convergence at all. Furthermore, we were able to show that for any value of λ between 0.1 and 0.5, that is $0.1 \leq \lambda \leq 0.5$ the exact solutions are attained, but the number of iterations before convergence vary. However, for any value of $\lambda \geq 0.6$ only the exact solution of y is attained but x will not show any sign of convergence.

Chosen initial points x = 1, y = 5 and $\lambda = \frac{1}{2}$ as discussed in [1], we have at 46th and 48th iteration 0.454738996489 and 8.75216885 for the values of x and y respectively which is not closed to the exact solution.

Itera.	x_i	y_i
0	1.0000000000000	5.000000000000
1	0.687500000000	5.12500000000
2	0.474376449025	5.25899638336
3	0.474727408840	5.39491189889
4	0.472363275981	5.52692538919
5	0.471492510909	5.65519074636
6	0.470033063863	5.77979091716
7	0.469036920871	5.90083849700
8	0.467925477309	6.01842767050
9	0.467007654274	6.13265825938
10	0.466093526565	6.24362320579
11	0.465285500464	6.35141540505
12	0.464511314049	6.45612392755
25	0.458245150792	7.57335363642
30	0.456957101889	7.90239239727
40	0.455330468724	8.43274734746
45	0.454823291377	8.64531096666
46	0.454738996489	8.68423190179
47	0.454659630855	8.72203494239
48	0.454584890208	8.75875216885
49	0.454514504642	8.79441474342
50	0.454448208085	8.82905293623
60	0.453963484049	9.12516412368
$\vdots$	$\vdots$	$\vdots$

Table 1: $x_0 = 1$, $y_o = 5$ and $\lambda = \dfrac{1}{2}$

Itera.	x_i	y_i
0	1.0000000000000	5.000000000000
1	0.375000000000	5.25000000000
2	0.722773408439	5.52422680412
3	0.226446813850	5.76798762068
4	1.36639047136	6.00859745247
5	0.676908564285	6.18456693603
6	0.200570018032	6.39487965638
7	1.55635430704	6.59842918752
8	0.878453256992	6.73892741142
9	0.179230384592	6.90855042595
10	1.72837178559	7.08189419085
11	1.07726874945	7.19566314280
12	0.298301189243	7.33227518516
25	0.897234655270	8.63548945304
30	0.277233908095	8.91632603683
40	2.40075310299	9.34201687328
41	1.88971882031	9.36284369384
42	1.23631135239	9.38641097307
43	0.403088863981	9.41470963621
44	0.611490879955	9.44847174074
45	0.156477780016	9.47934788320
46	1.99660351205	9.50829007054
47	1.37450321854	9.52586990187
48	0.567172457569	9.54671436573
49	0.213458203155	9.57232424146
50	1.57429797860	9.59655961073
60	0.232781839894	9.74314513280
$\vdots$	$\vdots$	$\vdots$

Table 2: $x_0 = 1$, $y_o = 5$ and $\lambda = 1$

Itera.	x_i	y_i
0	0.950000000000	9.800000000000
1	0.540604968337	9.80508344620
2	0.445754382501	9.80877224001
3	0.451620057508	9.81097181243
4	0.452459804156	9.81232996890
5	0.452730342193	9.81315968330
6	0.452848784708	9.81367103140
7	0.452908685552	9.81398599594
8	0.452941571381	9.81418037100
9	0.452960488181	9.81430036451
10	0.452971679969	9.81437447914
11	0.452978414012	9.81442026555
12	0.452982507933	9.81444855626
25	0.452989012007	9.81449423236
30	0.452989023273	9.81449431227
40	0.452989024380	9.81449432012
41	0.452989024383	9.81449432015
42	0.452989024386	9.81449432016
43	0.452989024387	9.81449432017
44	0.452989024388	9.81449432018
45	0.452989024388	9.81449432018
46	0.452989024389	9.81449432018
47	0.452989024389	9.81449432018
48	0.452989024389	9.81449432019
49	0.452989024389	9.81449432019
50	0.452989024389	9.81449432019
60	0.452989024389	9.81449432019
$\vdots$	$\vdots$	$\vdots$

Table 3: $x_0 = 0.95$, $y_o = 9.8$ and $\lambda = \dfrac{1}{F_n}, n = 2, 3, \ldots$

Itera.	x_i	y_i
0	1.000000000000	7.000000000000
1	0.620833333333	7.07500000000
2	0.480033894508	7.12946993363
3	0.466631737840	7.16236408114
4	0.463750389467	7.18271989426
5	0.462708473416	7.19516069833
6	0.462224289853	7.20282901870
7	0.461970598626	7.20755263529
8	0.461828271819	7.21046783862
9	0.461745298775	7.21226751590
10	0.461695801051	7.21337910885
11	0.461665865469	7.21406583168
12	0.461647608505	7.21449014868
25	0.461618511430	7.21517522268
30	0.461618460933	7.21517642115
40	0.461618455969	7.21517653896
41	0.461618455954	7.21517653933
42	0.461618455944	7.21517653956
43	0.461618455938	7.21517653970
44	0.461618455935	7.21517653979
45	0.461618455932	7.21517653984
46	0.461618455931	7.21517653987
47	0.461618455930	7.21517653989
48	0.461618455929	7.21517653991
49	0.461618455929	7.21517653991
50	0.461618455929	7.21517653992
60	0.461618455929	7.21517653993
$\vdots$	$\vdots$	$\vdots$

Table 4: $x_0 = 1$, $y_o = 7$ and $\lambda = \dfrac{1}{F_n}, n = 2, 3, \ldots$

Finally, by taking $\lambda = \frac{1}{F_n}, n = 2, 3, \ldots$ with x = 0.95 and y = 9.8, the convergence is attained at 46th and 48th iteration to obtained 0.452989024389 and 9.81449432019 for the values of x and y respectively. Chosen initial points x = 1, y = 5 and $\lambda = \frac{1}{2}$, we have at 46th and 48th iteration 0.454738996489 and 8.75216885 for x and y respectively.

References

[1] Russel, David, L. (1970): *Optimization Theory*. New York: W.A. Benjamin,Inc.

[2] Ibiejugba, M. A. (1985): *Computational methods in Optimization*. Ph.D Thesis, University of Leeds, Leeds, U.K. McGraw- Hill Book Co., Singapore.

[3] S. S. Rao (1996): *Theory and Practice of Optimization,third edition*:A Wiley-Inter science Publication John Wiley and Sons, Inc. New York / Chi chester / Brisbane / Toronto / Singapore.

[4] T. V. Mourik (2005): *Fortran 90/95 Programming Manual,fifth revision*: Chemistry Department, University College London.

[5] W. E. Mayo and M. Cwiakala (1995): *Theory and problems of Programming with Fortran 77*: McGraw-Hill Companies, Inc. U.S.A.

[6] E. K. Chong and H. Z. Stainslaw (2001): *An Introduction to Optimization*: 2nd Edition, A Wiley-Inter science publication. John Wiley and sons, Inc. New York/Chichester/Weinhein/Brisbabe/Singapore/Toronto

[7] J. W. Chinneck (2008): *Practical Optimization*: A Gentle Introduction. http://www.sce.carelton.ca/ffaculty/chinneck/po.html

[8] J. O. Omolehin (2006): *Further Results on CGM Algorithm*: ABACUS 33 374-388. Nigeria, Mathematics Association of Nigeria.

[9] S. Soft (1995): *Fortran 90 Users Guide*: A Sun Microsystems, Inc. Business,2550 Garcia Avenue Mountain View, CA 94043, U.S.A.

[10] J. O. Omolehin and K. Rauf (2006): *Computational Capability of CGM Algorithm*: Creative Math. 15 41-46 Department of Mathematics and Computer Science, North University of BAIAMARE Romania. http://creative-mathematics.ubm.rol.