Journal of the Nigerian Association of Mathematical Physics
Volume **12 (May, 2008), 409 -412**
© J. of NAMP

Asymptotic time complexity of an algorithm for generating the coefficients of the Chebyshev polynomials for the Tau numerical method

Dele Oluwade ** and O.A. Taiwo*
****Department of Computer Science, University of Ibadan, Ibadan, Nigeria.**
***Department of Mathematics, University of Ilorin, Ilorin, Nigeria.**
***email: oataiwo2002@yahoo.com, Phone no. 08067328211**

Abstract

We describe the asymptotic behaviors of an algorithm for generating the coefficients of the Chebyshev polynomials for the Tau numerical method, due to the roles it plays in obtaining the approximate solutions of both ordinary and partial differential equations. This algorithm when well implemented will reduce the problem of over- determinate always encountered by Tau-method.

Keywords**:** Asymptotic behaviour, Tau-method, algorithm
1991 Mathematics subject classification 68Q25, 65LO6

1.0 Introduction

In an earlier paper, Adeniyi, Olugbara and Taiwo [1], presented five generic algorithms for solving ordinary differential equations using the Tau numerical method. In the present paper, we discus the asymptotic time complexity of one of the algorithms, namely, the Coefficient of Chebyshev polynomial generator (CCPG) algorithm because of the role the Chebyshev Polynomial plays in using the Tau method to solve problems in ordinary and partial differential equations.

The Tau method, due to Lanzcos [9] is a popular technique for getting the approximate polynomial solution of linear ordinary differential equations with polynomial coefficients. For the purpose of our discussion, we shall consider the math order linear ordinary differential equation given by.

$$LY(x) = \sum_{i=0}^{m} P_i \ (x) Y_n^i (x) = f(x) \tag{1.1}$$

where,

$$P_i(x) = \sum_{i=0}^{N_i} p_{ij} x^i, \ f(x) = \sum_{i=0}^{F} f_i x^i, \ x\varepsilon \left[a, b\right] \tag{1.2}$$

Here N_i , F are given non-negative integers, and f_i, P_{ij} are given real numbers. The smooth solution $Y(x)$ satisfies a set of multi-point boundary conditions given by

$$L^* Y(x) = \sum_{i=0}^{m-1} a_{ik} Y^{(I)}(x_{ik}) = \beta_k, \ k = 1(1)m \tag{1.3}$$

where α_{ik} , x_{ik} , β_k are given real number $x_{ik} \varepsilon [a, b]$ such that (1.3) is satisfied. $Y^{/}(x)$ and $Y^{(0)}(x)$ are respectively defined by

$$Y^{(?)}(x) = \tfrac{d}{dx} Y(x) \tag{1.4}$$

$$Y^{(0)}(x) = Y(x) \tag{1.5}$$

**email: doluwade@mail.scannet.com

and L is the operator defined by $$L = \frac{d}{dx^2} + \frac{d}{dx} + 1 \qquad (1.6)$$

To solve (1.1) - (1.3) using the Tau method, we seek an approximate solution of the n-th order degree polynomial $Y_n(x)$ given by $$Y_n(x) = \sum_{i=0}^{n} a_i x^i, \; n \pi \infty \qquad (1.7)$$

This polynomial is called a Tau approximant. That is, we construct the economized polynomial Y_n(x) which is the exact solution of the perturbed problem (otherwise called Tau problem) given by

$$LY_N(x) = \sum_{i=0}^{n} P_i(x) Y_n^{(i)}(x) = \sum_{i=0}^{F} f_i x^i + H_n(x) \qquad (1.8)$$

$$L * Y_n = \sum_{i=1}^{m-1} a_{ik} Y_n^{(i)}(x) = \beta_{k,} \; k = 1(1m) \qquad (1.9)$$

Economization of power series essentially involves the reduction of a given polynomial of degree n to one of degree *n* -1. The perturbation term H_n (x) in (1.8) is defined by

$$H_n(x) = \sum_{i=0}^{m+s-1} \tau_{m+s-1} T_{n-m+i+1}(x) \qquad (1.10)$$

Where $\tau_{1,} \tau_{2,} \Lambda \; \tau_{m+s}$ are the Tau parameter, s is the number of over determination of (1.1), given by *s* = max(N_i -1) (for i∈ $[0, m]$, F$\leq m + s$ and $T_r(x), \; r = n - m + 1, \Lambda \; n + s$ are the *r*-th degree Chebyshev polynomial is defined in the interval [*a, b*] by

$$T_r(x) = Cos\left(rCos^{-1}\left(\left\{\frac{2(x-a)}{b-a}\right\} - 1\right)\right)$$

By letting $T_r(x) = \sum_{i=0}^{r} C_i^{(r)} x^i$, we have $C_i^{(r)} = 2^{2r-1}(b-a)^{-r}$, $T_i, \; i = 1,2,\Lambda \; , m+s$, are fixed parameters which make (1.9) to be satisfied exactly. The error equation is defined by $Le_n(x) = -H_n(x)$ where the global error function e_n (x) is given by $e_n(x) = \max|y(x) - Yn(x)|$, $a \leq x \leq b$.

2.0 The Chebyshev polynomials

The Chebyshev polynomials belong to the class of polynomials known as orthogonal polynomials. Other polynomials in this class include Legendre polynomials, Laguerre polynomials and Hermite polynomials. There are different kinds of Chebyshev polynomials, each of which is defined within a specified interval. These include [8]

(a) Chebyshev polynomials of the first kind.
(b) Modified Chebyshev polynomials of the first kind.
(c) Chebyshev polynomials of the second kind.
(d) Modified Chebyshev polynomials of the second kind.

In particular, the Chebyshev polynomials of the first kind T_r (x) are defined in the interval [-1, 1]. This class of polynomials satisfies the following properties, amongst others (see refs [6, 7])

(i) The zeros of the polynomials are given by $x_k^{(r)} = Cos\frac{2k-1}{2r}\pi$, where *k* = 1, 2,…,*n*

(ii) The leading term in $T_r(x)$ is $2^{r-1} \; x^i$

(iii) $T_r(x)$ is irreducible over Q only if n = 2^k, k = 0, 1, 2,.....where Q is the set of rational numbers.

(iv) $T_r(x)$ possesses the semi-group property i.e. $T_r(T_s(x)) = T_{rs}(x)$

(v) $T_r(x)$ is commutative, i.e $T_s(T_r) = T_r(T_s)$

(vi) Only the Chebyshev polynomials of the first kind can commute with a given $T_r(x)$ if $n \geq 2$

3.0 Time complexity

In this section, we examine the efficiency of the CCPG algorithm. The efficiency of an algorithm is typically a reflection of the time and space complexities of the algorithm. While the time complexity is measured in seconds of computer time, the space complexity is measured by the number of bytes of computer memory. Both complexities, in general, depend on the input data and the size of the input data. Also, both complexities depend on the particular computer which is being used. Some of the characteristics of a computer which affect the time and space complexities include:
(a) Instruction data, (b) Speed of the hardware, (c) General architectural make-up and (d) Type of computer (or translator) used in the execution of the program.

Asymptotic complexity refers to the behaviour of an algorithm when the size of the input gets very large. In general, it is the time complexity which is normally of great interest in the design and analysis of algorithm. This is because memory is normally not in short supply.

3.1 The coefficients of Chebyshev polynomial generator (CCPG) algorithm

The input (arguments) of the algorithm is the lowest and highest degrees of the Chebyshev polynomial (NLO and NGS respectively) and the lower and upper limits of interval (A1 and B1 respectively) within which a given problem is defined. C1 and C2 are vectors used in the generation of Chebyshev polynomials, whilst F1 and F2 are auxiliary variables. NDG, NG1, I, J, and K are loop counters.

The CCPG algorithm in C-language format is as follows [1]

```
1.      Begin
2.      C1 [0] = 1
3.      C2 [0] = (B1+A1) / (B1-A1),
4,      C2 [1] = 2 / (B1-A1),
5.      F1 = 4 / (B1-A1),
6.       F2 = 2* C2 [0],
7.      For (NDG = NLO, NDG < = NGS, NDG ++)
        {
        NG1 = NDG + 1
        If (NG1 = = 1) C [0] = C1 [0]
        else if (NG1 = = 2)
        {
        C [0] = C2 [0]
        C [1] = C2 [1]
        }
        else
        {
        C [0] = F2* C2 [0] - C1 [0]
        For {K = 3; K < = ng1; K++)
        {
8.      For {I = 1; 1 < K ; I ++
        C [1] = F1* C2 [1-1] + F2*C2 [1] - C1 [1]
9.      For (1 = 0; I < K, 1 ++)
```

```
C1 [1] = C2 [1];
C2 [1] = C [1]
}
}
}
```

```
10.     For (J = 0; J < NG1; T [NDG] [J] = C [J], J++);
        }
11.     END
```

***Theorem* 3.1**

Let $f_1(n) = 0(g(n)$ *and* $f_2(n) = 0(g_2)(n))$. *Then,*

(i) $f_1(n) + f_2(n) = 0(g_1(n) + g_2(n))$

(ii) $f_1(n) + f_2(n) = 0(\max(g_1(n), g_2(n)))$

(iii) $f_1(n) \cdot f_2(n) = 0(g_1(n) \cdot g_2(n))$

where 0 *is the asymptotic upper bound notation which may or may not be asymptotic tight.*

Proof:

(i) Since $f_1(n) + f_2(n) = 0(g_1(n))$, then $\exists C_1, n_1$ such that $0 \le f_{1(}(n) \le C_1 g_1(n)$. Similarly, $\exists C_2 n_2$ such that $0 \le f_2(n) \le C_2 g_2(n)$. Let $C_3 = C_1 + C_2$, $n_3 = n_1 + n_2$ and g_2 *(n) =* g_1 *(n) +* g_2 *(*n_1 *+* n_2*),* f_1 $(n) \le C_1 g_1(n)$ $\forall \cap \ge n_3$, $f_1(n) \le C_3 g_1(n)$, $\forall \cap n_3$ and $f_1(n) \le C_3 g_3(n)$, $\forall \cap n_3$ also, $f_2(n) \le C_3 g_3(n)$ $\forall \cap n_3$. Hence, f_1(n)+ $f_2(n)$ $\le 2C_3 g_3(n) \le (n)$. And the result follows. Thus (ii) and (iii) can similarly be proved

***Theorem* 3.2**

The asymptotic time complexity of the CCPG algorithm is of linear time.

***Proof*:**

Let T (n) be the asymptotic time complexity of the CCPG algorithm. Let $T_i(n)$ be the time complexity of step i of the algorithm. We note that the algorithm has 11 steps altogether.

$T_j(n) = 0(1)$ $\forall 2 \le j \le 6$, $T_7(n) = (n - r + 1)$ 0(1) = 0($n - r$ + 1), $T_4(n) = (n-1)$ 0(1) = 0(n - 1)

$T_9(n) = n$ 0(1) = 0(n), $T_{10}(n) = n$ 0(1) = 0(n), T(n)= $\sum_{i=2}^{10} T_i(n) = 0(n)$ and so the theorem is proved.

4.0 Discussion and conclusion

We presented the generic algorithm for Coefficient of Chebyshev Polynomial Generator (CCPG) which could be used in the numerical solution of ordinary and partial differential equation via the Tau method. The purpose of the algorithm developed serves as a powerful tool to generate the coefficient of Chebyshev polynomial which required toe generated at all time as the computation processes.

We also studied the behaviour of the algorithm as the size of the input gets very large. The inputs of the algorithm are the lowest and the highest degrees of the Chebyshev polynomial as well as the lower and upper limits of interval with which a given problem is defined and the algorithm is easily used at different intervals.

The authors showed the asymptotic time complexity is of linear time. Hence the algorithm is asymptotic efficient.

Reference

[1] Adeniyi R.B, Olugbara, O.O. and Taiwo, O.A (1999), "Generic algorithms for solving O.D.E using the Tau method with an error estimation" International Journal of Computer Mathematics, 72, 6, 63 – 80.

[2] Adeniyi R.B, Onumanyi, P. and Taiwo, O.A. (1989)."A computational error estimate of the Tau method applied to some non-linear ordinary differential equations. ABACUS, Journal of the Mathematical Association of Nigeria. 19(2), 99-111.

[3] Azmoodeh, Manoochehr (1990). Abstract Data Types and Algorithms Macmillan Education Ltd.

[4] Cormen, Thomas H., Leiserson, Charles, E. and Rivest, Ronbald, L (1990), Introduction to Algorithms. The MIT Press, Cambridge, Massachusetts.

[5] Parsons, Thomas W. (1995), Introduction to Algorithms in Pascal, John Wiley & Sons Inc, New York.

[6] Churchhouse, Robert, F. (ed) (1981). Handbook of Application Mathematics Vol. III Numerical Methods. John Wiley & Sons Inc, New York.

[7] Rivilin, Theodore, J. (1990), Chebyshev Polynomials (from approximation theory to algebra and number theory) John Wiley & Sons Inc, New York.

[8] Lyusternik, L.A. Chervonenkis, O.A & Yanpol'skii, A.R (1965). Handbook for computing elementary functions. (Trans Tee, H. J. Trans (Ed) Stewart, K. L.) Pergamon Press, Oxford.

[9] Lanczos, C. (1938), Trigonometric Interpretation of empirical and analytical functions" Journal of mathematical and physics Vol 17,123-199.